Mobile Transfer Client™ for EFT™ v6.5.16

ADMINISTRATION GUIDE

globalscape™
securely connected

GlobalSCAPE, Inc. (GSB)

Address:	4500 Lockhill-Selma Road, Suite 150 San Antonio, TX (USA) 78249
Sales:	(210) 308-8267
Sales (Toll Free):	(800) 290-5054
Technical Support:	(210) 366-3993

Web Support: http://www.globalscape.com/support/

Last updated: February 19, 2014

Table of Contents

(This page left blank for 2-sided "book" printing.)

Mobile Transfer Client (MTC)

The Mobile Transfer Client (MTC) application (app) provides a way for your iOS and Android phone and tablet users to securely connect to EFT and upload and download files while providing a number of centrally managed security controls for safeguarding your corporate data.

Mobile Transfer Client Features

EFT's Mobile Transfer Client supports the following features:

Security

- Secure communications and transport over HTTPS
- SSL certificate management (accept CA-signed certificates, otherwise prompt)
- Secure data storage
- Central policy management that controls:
    - Profile password storage
    - Data caching
    - Storing data in an offline repository (vault)
    - Sharing files via email
    - Opening files in external (third-party) apps

Profile Management

- Multiple profile support
- Single "tap-on" link for automatic profile provisioning
- Dual-stack (IPv4 and IPv6) support
- International Domain Name (IDN) and Punycode support
- Support for non-default ports
- Auto-login to last connected profile on app launch
- Password reset and recover lost username support
- Full support for Unicode characters

Files Listings and Transfers

- View up to 10,000 files in a directory listing
- Transfer files up to 3GB in size
- Transfer multiple files concurrently
- Pause and resume transfers
- Automatic resume of system paused transfers
- Resume partial transfers from point of failure
- Download files to a separate secure repository for offline access
- Download files and open them using the built-in file viewer (Only certain file types are supported.)
- Open text, log, and other ASCII files in the internal text viewer
- Download files then open in an external program
- Download files and share them as email attachments
- Download files, make edits, then upload the modified version
- Automatic and transparent file integrity checking
- Create, rename, and delete folders
- View download progress
- Abort transfers and retry failed transfers

General Settings and Logs

- Clear profile and vault caches
- Specify the maximum cache size
- Enable logging, including verbose logging
- View detailed transaction logs
- Email logs to your administrator
- Clear all logs
- Disallow password saving (global option)
- Custom/branded profile icons (optional)

MTC System Requirements

MTC is supported on Android- or iOS-based mobile devices of varying resolutions.

- EFT v6.5.16 and later, Standard or Enterprise
- Android 2.3 or later for general operations
- Android 3.0 or later if encrypted data store is required
- iOS 6.1 or later (tested on both 6 and 7)

Mobile Transfer Client Licensing

Users with accounts on EFT can use the Mobile Transfer Client (MTC) to connect to EFT during the EFT trial period or if the MTC module has been activated (registered), assuming MTC access is enabled for a particular Site. Users will receive a “503 forbidden” message if they attempt to connect with MTC past the trial period or if MTC has not yet been activated.

To activate the MTC module

- Refer to Activating EFT and Modules.

Enabling the Mobile Transfer Client and Configuring Security

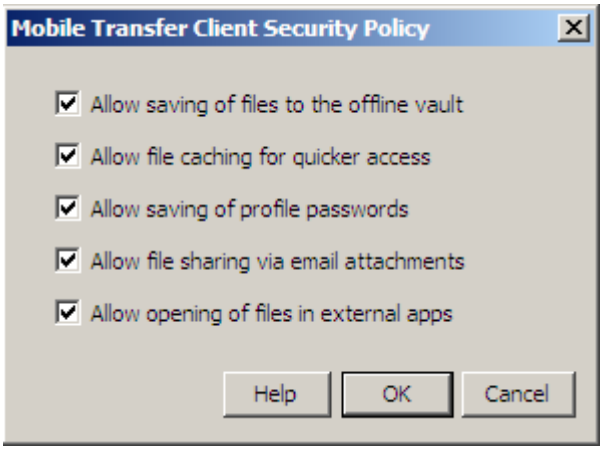
Perhaps the most important feature available to the MTC, aside from an always secure connection, is the centrally managed security controls that dictate what users can and cannot do within the MTC app when connected to EFT.

Server and Site administrators can also block MTC connections, effectively terminating the connection based on the user-agent string that identifies the client as an MTC client. (This does not prevent other file transfer clients—mobile or desktop—from connecting to EFT.)

MTC's security policy only applies to files in the remote directory. The security policy does not apply to files in the vault, which means that any file downloaded to the vault can be shared or opened in third-party applications. If the EFT administrator doesn't want users to share files or open them in third-party apps, then EFT should be configured to not allow users to save files to the vault.

To enable/disable the MTC and configure security controls

1. In the administration interface, connect to EFT and click the **Server** tab.
2. On the **Server** tab, click the Site that you want to configure.
3. In the right pane, click the **Connections** tab.
4. If not already enabled, enable HTTPS and create and assign an SSL certificate for the Site.
5. Select the **Allow Globalscape Mobile Transfer Client (MTC) over HTTPS** check box.
6. Click **Configure**. The **Mobile Transfer Client Security Policy** dialog box appears.



7. Each time a user connects or makes specific requests to MTC, their profile is updated with the latest security control settings. Select (enable) or clear (disable) the following check boxes:
    - **Allow saving of files to the offline vault**—The vault in the MTC app is an encrypted storage area* where a user can download a *copy* of a remote file (from this or possibly another EFT account) for subsequent access, even when offline, and even if the user no longer has an account on EFT. If you disable the ability to store files in the vault then you should also consider disabling **Allow file sharing via email attachments** and **Allow opening of file in external apps**, since in all three instances files essentially leave your control.

- **Allow file caching for quicker access**—The cache is an account (MTC profile)-specific secure storage area* that MTC uses to keep copies of files that were downloaded. The next time a user taps on a file (to open in the internal PDF viewer, for example), the file will be opened from the device's cache (assuming it isn't stale data) rather than downloaded again from the server, resulting in a better end-user experience. The cache is semi-permanent in that it will grow as files are downloaded, and is not cleaned up unless space is needed or the user decides to clear the local cache (similar to how a browser's cache works). Disabling this option doesn't disable file caching altogether, but rather makes caching temporary, cleaning up cached data upon application exit.
- **Allow saving of profile passwords**—Preventing the user from saving the password forces them to re-type it each time they connect to EFT. Once authenticated, the app will retain the password in memory until the app is exited. If this setting is disabled in EFT, the password field is grayed out in the MTC app (for this profile) and passwords for this profile are removed from the mobile keychain (iOS) or database (Android), if stored there. If this setting is enabled, then it is up to the user to decide whether to store the password in the MTC app.
- **Allow file sharing via email attachments**—After downloading a file, an MTC user can optionally share the file as an email attachment with another user (taking the file outside of MTC's control). If this setting is disabled in EFT, then sharing won't be allowed from within the MTC application. If you disable the ability to share files, then you should also consider disabling **Allow opening of file in external apps** and **Allow saving of files to the offline vault**, since in all three instances files essentially leave your control.
- **Allow opening of file in external apps**—After downloading a file, an MTC user can optionally open the file in a third-party app, which is often necessary if there is no internal MTC viewer that can open the particular file extension. When the user performs an "Open In" operation, the file is decrypted so that the external app can accept the file (which is now outside of MTC's control). This setting provides administrators the ability to block users from opening files in other apps, forcing them to use the in-app viewer (if available) or nothing at all. If you disable the ability to open files, then you should also consider disabling **Allow file sharing via email attachments** and **Allow saving of files to the offline vault**, since in all three instances files essentially leave your control.

  *Read the section on data encryption in the MTC FAQ. Also keep in mind that files saved to disk on Android are not sandboxed as in iOS, thus minimizing the effectiveness of some of the above controls, although MTC does leverage Android's "Internal Storage" for keeping it segmented and inaccessible to other apps, at the cost of much less available disk space than if the standard physical storage had been used, which is "world shareable" and thus unsuitable for storing corporate data, even if device encryption is enabled. (The third-party app cannot arbitrarily do so; the user must perform an "Open" operation from the other app and select the file located elsewhere in the data folder.)

8. Click **Apply** to save the changes on EFT.

To disable MTC/block MTC connections

- Clear the **Enable Globalscape Mobile Transfer Client (MTC) over HTTPS** check box.

  - or-

- Disable HTTPS. (The MTC requires HTTPS.)

Onboarding Mobile Transfer Client Users

Typing a host address, username, and complex password on a mobile keyboard can be frustrating. To make the onboarding process easier, EFT generates a single-click hyperlink each time a user is created (and when the password is reset) and includes that link in the welcome email generated by EFT and sent to the user. This link includes the information necessary to connect to EFT (host address, port, path, username, and password) in encoded format. When the user receives this link and taps (clicks) the link in their mobile or tablet device's email client, the MTC app is launched (if installed) and automatically provisioned, giving the user access to files.

The link sent from EFT is encoded, but not encrypted. If you don't want user passwords to be communicated via email, then modify EFT's settings to only send the username. You will need to find another way communicate the user's password.

To properly onboard users

- Make sure EFT's SMTP settings are properly configured.
- Under the Site's **Security** tab you must select the **Enable option to e-mail users their login credentials** check box.
- When you create a user (or change their password) you must select the check box **E-mail login credentials to user**.
- The credentials template file (`CredentialsEmail.tpl`) must include the `#MTC_URL` section and the `MTC_LINK` variable, which are present by default.
- The user must have the app installed prior to tapping on the MTC link. (The instructions in the welcome email will include this link.) When the user taps the link, the mobile operating system will associate the link with the MTC app, because it is registered to that particular link format.
    - If MTC is not found, the operating system will display an error.
    - If MTC is present, it will launch, decode the parameters, create a profile in MTC with those parameters, immediately connect to EFT, and display the user's directory listing.

Decommissioning Mobile Transfer Client Users

User account deactivation is the same for Mobile Transfer Client (MTC) users as for any other user account in EFT. However, there is one more step you can take that will result in removing that decommissioned user's cached data, effectively wiping your corporate data off the device, insofar as MTC's data repository is concerned. Use one of the methods below to clear the user data.

Method 1

Before deactivating the user, delete all the files in the user's home directory, including sub-folders. Upon subsequent login, MTC will synchronize with a now empty directory, effectively wiping any cached files stored in that profile's repository. Once you have verified that the user has completed their final login (perhaps through an Event Rule notification), you can disable or remove their account to prevent further logins.

Method 2

If deleting the user's data is simply not an option, then disable the **Allow file caching for quicker access** option in MTC's security policy configuration. Upon next login, that user (as well as all other users) will receive the new policy, and next time they close their MTC app, their cached data for that profile will be deleted. As in method 1, you will want to disable or delete their account after that final authentication. It is up to you to decide whether to re-enable the **Allow file caching for quicker access** option or just keep it turned off, which is arguably more secure, but less user friendly for your mobile users.

Custom Branding of the Mobile Transfer Client Profile

When users connect to an EFT Site using the Mobile Transfer Client (MTC), EFT can *optionally* deliver a small graphic file that will replace the generic icon shown in the MTC app for that profile in the Profile address book. You can place the custom graphic in the Server folder for use on every Site on which the MTC is enabled, or you can place custom graphics in individual Site-specific folders if you have different languages on different Sites (for example).

To provide a custom logo or graphic

1. Create a .png file that is 200x200 pixels.
    - The file can include alpha transparency.
    - Image squares larger than 200x200 will be scaled down, but will use more bandwidth. Non-square images or square images smaller than 200x200px will be rejected by the client.
2. Save the file as **icon.png**.
3. On the EFT computer, do one of the following:
    - To use the icon Server wide, copy **icon.png** to the **\mtc** directory (e.g., **C:\Program Files (x86)\Globalscape\EFT Server Enterprise\web\public\EFTClient\mtc**).
    - To deliver a Site-specific icon (as opposed to Server wide), then place the icon under **\web\custom\[SiteName]\EFTClient\mtc**.

You can create a different custom MTC folder for each Site.

To customize files on a Site

1. In the **\web\custom** folder, create a directory structure in the form **[SiteName]\EFTClient\mtc**.
2. Copy only the default files that you want to edit (rebrand) from the **\web\public\EFTClient\mtc** folder into the **\web\custom\SiteName\EFTClient\mtc** folder that you created. (It is not necessary to copy all of the default files.)
3. Edit the *copy* of the file in the **\custom\SiteName\EFTClient\mtc** folder, and save it.

When upgrading, the **\custom** and **\public** folders are backed up and renamed with the date and time (e.g., **\customBackup_9-28-2010_16-18** and **\publicBackup_9-28-2010_16-18**).

Upon initial installation, this **\custom** directory is empty. You must create the directory structure for any Server (**\custom\EFTClient**) or Site (**\custom\MySite\EFTClient**) branded files. If you have multiple Sites, each Site can have different branding (e.g., one can be in English and one in French). EFT first looks in the Site's custom (branded) directory **\web\custom\MySite\EFTClient** and loads any branded files. For files that are not present in the Site's **\custom** directory, EFT checks the Server's **\custom** directory, **web\custom\EFTClient**, and then loads the files that it finds there. Finally, for any other files, it will load the default files from **\web\public\EFTClient**. Branded files that are Site-specific override any Server-wide branded and default files, while branded files that are Server-wide override the default (Globalscape-branded) files provided by the installer.

- The best practice is to have only customized files in the **\custom** folder and to leave the default files *unmodified* in the **\web\public\EFTClient** folder.
- The Site folder **\web\custom\[SiteName]\EFTClient** should hold just those files that contain customizations for that Site.
- The Server folder **\web\custom\EFTClient** should hold just those files that contain customizations for the Server.
- The Server-branded files will apply to all Sites defined on the Server, but any Site-branded files will override the Server-branded files.
- It is not necessary to restart the Site or Server to see your changes, but you will have to refresh or close and reopen your browser.

Obtaining the Mobile Transfer Client Apps

The Mobile Transfer Client (MTC) is available for both Android and iOS devices as an app, also called a native app (as opposed to a browser-based app).

To download the app onto a device

- Go to the iTunes or Google Play store on your device, and search the app store for "Mobile Transfer Client by Globalscape, Inc." The links to the MTC app are also included in the credentials template email.

Mobile Transfer Client FAQ

Frequently asked questions (FAQ) are answered below.

Does MTC require a certain version of EFT Server?

Yes, MTC will only connect to EFT versions that support the *mobilepolicysettings* web service call in EFT v6.5.16 and later, Standard or Enterprise.

What protocols does MTC support?

MTC only uses HTTPS. This protocol provides transport security and a rich mechanism (using headers) for communicating with EFT about things like security policies, file checksums, and other advanced features that older protocols such as FTP and SFTP cannot provide. On EFT Standard, the HTTPS module is required.

What if I want to use FTP or SFTP?

There are plenty of free and for-pay FTP and SFTP clients available to iOS and Android; however those apps do not offer the same policy and security controls as provided by MTC.

What authentication mode does MTC use?

MTC relies on Basic Auth over an encrypted HTTPS connection. Session (Form)-based authentication is being considered for a future version.

Why can’t I just use my browser to download files instead of your native app?

You can to a certain extent if you connect to EFT and bypass the Java-based client option. However, what you can do once you connect is severely limited by the mobile operating system and browser you choose to use, and also lacks the security policy features provided by MTC.

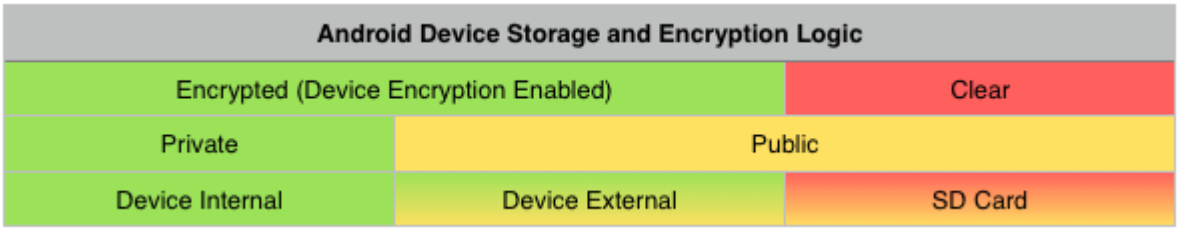
What prevents my users from using a mobile SFTP or FTPS client or third-party browser?

Nothing really; but that is no different from today when it comes to your user’s choice of desktop-based client. As the administrator you can turn off SFTP or FTPS support entirely, or allow these protocols knowing that you can’t control the app your user chooses to use as their client. The benefit of using MTC is that you can set a corporate policy that mandates that users only use the MTC client to interact with EFT from their mobile device, as MTC provides a level of governance and control due to its centrally managed security policies. EFT logs can demonstrate whether users are using MTC or not based on the protocol used and the http user-agent string (keep in mind user-agent strings can be forged).

Does the MTC app protect (encrypt) data at rest?

Yes. MTC leverages the OS level encryption for encrypting contents at rest. In iOS this means the user MUST be using a pin code to unlock their device. If the user has not established a pin code then data will not be encrypted while at rest. When the device is unlocked (user enters their pin code), an OS-wide decryption key is created that MTC will leverage when reading files from disk. When a user performs an "Open In" or "Share as Link" operation, MTC takes a decrypted COPY of the cached file and passes it to the third-party app, assuming those operations are allowed by the security policy. The third-party app may or may not use the encryption class, meaning the file is not guaranteed to be encrypted when saved to disk by the third-party app (e.g. they assign the file to the NSFileProtectionNone class).

Android is a bit more problematic as apps and their data are not completely sandboxed. Starting with Android 3.0 you can enable whole disk encryption; however, once you enter your pin code, your device is decrypted device-wide, and nothing prevents the user or another app on the device (deliberately run by the user or otherwise) from accessing the data directory for any other app. To prevent this breach of data privacy, MTC leverages Android's so-called "internal" data storage, a relatively small partition of the overall non-removable physical storage that acts like a data sandbox, preventing both the user and app from accessing files downloaded from EFT into MTC's cache or offline vault. Contrast this "internal" storage with the so-called "external" storage (not to be confused with the SD card) which represents the remainder of the non-removable hard-drive and allows any app on the device to access any other apps' data. The downside of using internal storage is that it is usually about a tenth or less of the overall non-removable disk space. Using an SD for storage is simply out of the question because neither encryption nor data privacy are extended to this truly external, removable storage media type. Below is a graphic that helps illustrate Android's various partitions and how encryption and privacy (data sandboxing) apply. MTC follows the purely green path, assuming encryption is enabled device wide.



Are there any restrictions for MTC to work in EFT?

Yes. HTTPS must be enabled and the Site must rely on a single-factor authentication manager (AD, LDAP, ODBC, or GS Auth). Sites using RSA SecurID, RADIUS 2FA, or CAC-based authentication will result in failed MTC login attempts.

How do I troubleshoot MTC connection problems?

MTC can optionally keep a detailed log of all its transactions including the HTTP transport stream. There is even an option in the MTC log viewer for the user to email the log so that you can review the HTTP requests and responses and assist the user in determining why a certain operation failed. Alternatively, you can use EFT's eft.log file (after enabling the HTTP logger) to view decrypted HTTPS sessions. Most of the time what you are looking for is 401s (authentication failed), 404s (the resource requested was not found), or 503s (insufficient permission for the requested resource).

Does MTC support forced password reset, user-initiated password changing, lost password reset, and username recovery?

Forced password reset upon initial (or next) login is supported by MTC. User-initiated password change is not supported, but they can still do this from their desktop or even mobile browser. MTC also supports lost password reset and lost username recovery.

I have multiple Sites that are accessible to the same set of users. Can MTC accommodate multiple Sites?

Yes, MTC supports the concepts of Profiles, which are essentially the same as accounts on EFT Sites.

Can my users download a file to their device, make changes to the file, then re-upload the file back to EFT?

Yes, assuming "Allow opening of files in external apps" is enabled.

- On iOS devices, when the user chooses to open a file in a third-party app, MTC checks to see if a local cached copy exists and is fresh. If the file is not in cache or is stale, the file is downloaded from the server and then a copy of the file is passed to the third-party app. Once the user is done making changes in the third-party app (assuming the app can edit that file type) the user would then select the "Open In" or equivalent function in the third-party app and choose MTC as the destination. Files copied back into MTC are placed in the MTC offline vault. Back in MTC, the user can select files in their vault and upload those to EFT, effectively overwriting the original file. The download, open in third-party app, open back in MTC, then upload, are all separate operations.
- On Android devices, MTC takes on a more active role for file editing. When the user downloads the file, selects the **Open In** function, and then selects a third-party app, MTC passes a handle to the originally downloaded (and cached) file, rather than a copy of the same. MTC then spawns a file monitoring thread to keep track of changes (saves) made to that file. The user makes their edits in the third-party app, and after saving their changes, they must switch back to the MTC app. Once MTC is in the foreground, it cancels the monitoring thread. If changes were recorded, it immediately uploads the file to the server, overwriting the original file.

Does MTC leverage any Mobile Device Management (MDM) technology?

Globalscape does not rely on any third-party MDM technology to protect and manage MTC. Using an MDM provider would have significantly increased the cost and complexity, while providing only marginal benefits and in some cases drawbacks. For example, some MDM providers do not protect app data from being leaked to cloud services, a shortcoming that was deliberately addressed in MTC by restricting the user's ability to share or open files outside of the context of the MTC app. Other MDM providers protect network communications but don't leverage or enforce data encryption. Another major drawback of using MDM is that it requires an MDM agent be installed on the device, which is not always practical or possible when dealing with personal devices of users that might not be employees of your organization, but still need access to your data (e.g., customers and business partners).

Can the security policies for MTC be set on a per-user basis?

MTC's security policies must be set on a Site-wide basis, affecting all templates and users belonging to that Site. Please contact us if you feel that you need template-level or user-level control over MTC's security policies, so we can determine whether to extend more granular control over those policies in a future version release.

Can the security policy for blocking MTC from opening files be specific to certain file types?

When **Open In** is disallowed, it will affect all files, regardless of extension.