

IccRetrieveSecuredFile White Paper

Contents

Description.....	1
lccRetrieveSecuredFile	2
Installation.....	3
Providing The URL To An End User/Program.....	3
Application Configuration File.....	4
Logic File Description/Syntax.....	5
Logic File.....	5
Encrypt A File.....	18
Logic File Encrypted Values	19
Logic File Examples	19
Basic Retrieve Examples	19
Encrypted File Retrieve Example	20
Enable User Tools	21
Definitions	23
Modifications.....	23

Description

This document describes how to use the lccRetrieveSecuredFile.

lccRetrieveSecuredFile

is a website program to supply a secured file to a web user. It can supply the file as a downloadable attachment, or direct/display (i.e. in a browser).

20191024 update: New 'User Tools' module. Allows users to upload/encrypt files for others to download, as well as an 'invite' option.

Additional Capabilities

- set Valid Before/After dates
- set Maximum Downloads
- set Maximum Size

Program Logic

IT BackEnd Provided Files

- Reads the file the admin created Application Configuration File to find the Logic File path
- Reads the Logic File to find settings/Sets
- Compare the URL provided by a web viewer to the Sets
- If valid Id/Key found
 - Retrieve the secured file into memory
 - (optional) If key 'lcc:securedFileSetEncrypted' supplied, unencrypted file in memory
 - Downloads the file to the end user's browser using the 'show filename' provided by the Set setting

User Provided Files

- Requires LDAP Log-in, or using an 'Invite Link'
- Reads the file the admin created Application Configuration File to find the Logic File path
- Reads the Logic File to find settings/Sets
- Reads the individual Logic File for the download or invite
- If valid Id/Key found
 - Retrieve the secured file into memory or allow an Invitee to upload
 - Downloads the file to the end user's browser using the 'show filename' provided by the Set setting, or allows upload

Note: the IT BackEnd Provided Files and User Provided Files modes can be used from the same main Logic File/instance at the same time. In this case, it will serve both modules.

Program User Module Added - 20191024

This module allows users to upload their own files, which will then be auto encrypted, secured URL provided, or allow a user to 'invite' an outside user to upload.

Installation

Note: we recommend running the program on an Apache server. Though originally created for us in IIS, the IIS platform has introduced modules that make developing CGI program less stable. By running in Apache, a CGI can decide what 'it' wants to do, instead of the web server controlling the process/information being provided. This decision was made because of multiple changes/issues with IIS, including this one, where you have to create a fake configuration key/value to satisfy IIS built in engine expecting a CGI to act like a ASPX product, i.e. using a .config file/etc.

```
*erg* Solve HttpUtility.UrlDecode Configuration File not found error with  
AppDomain.CurrentDomain.SetData("APP_CONFIG_FILE", "...your fake config file path... ");  
my stackoverflow issue on this: https://stackoverflow.com/questions/46614944/iis-console-application-directoryentry-searchresult
```

- place the lccRetrieveSecuredFile.exe on a web server (recommend Apache)
- create/configure the lccRetrieveSecuredFile.exe.config (see Application Configuration File section)
- create/configure the Logic File (see Logic File section)
- (if logging supplied) set permissions on the Log folder to allow writing
- (if Max Downloads supplied) set permissions on the Max Downloads folder to allow writing

The User Toosl mode utilizes multiple CSS styles. We recommend using the CSS file include and place a link to it in the content provided by the key lcc:topperPath. See the 'topper' and 'footer' files for default/expected HTML code.

Providing The URL To An End User/Program

BackEnd Configured Downloads

To give access to a secured file, provide the URL, using:

Syntax: [https:\[your server\]/\[your program path\]?id=\[Set Id\]&key=\[Set Key\]](https://yourserver.com/yourprogrampath?id=[SetId]&key=[SetKey])

Example: <https://ourcollege.edu/scripts/lccRetrieveSecuredFile.exe?id=ourSecuredFile1&key=ourSecretKey>

User Uploads/Invites

To allow users to log-in and upload/send invites. This is done by adding "?mode=userTools" to the end of the URL.

Syntax: [https:\[your server\]/\[your program path\]?mode=userTools](https://yourserver/yourprogrampath?mode=userTools)

Example: <https://ourcollege.edu/scripts/lccRetrieveSecuredFile.exe?mode=userTools>

Application Configuration File

The program uses an Application Configuration file that must reside in the same location as the program and be called:

lccRetrieveSecuredFile.exe.config

The only purpose of this file is to load the Path of where the Logic File is located.

Change the "value" to the path of where the Logic File is located and named.

Security Recommendations (optional):

- store the actual Logic File this file points to in a non-web accessible location
- restrict permissions on the Application Configuration File to not be readable by the account being used for the website, only by the Application Pool account and/or restrict IP access to the local web server IP(s), including 127.0.0.1 local address.

File Syntax

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <appSettings>
    <add key="lcc:logicPath" value="..." />
  </appSettings>
</configuration>
```

File Example

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <appSettings>
    <add key="lcc:logicPath" value=" f:\folder\data\lccRetieveSecuredFile-logic.txt " />
  </appSettings>
```

</configuration>

Logic File Description/Syntax

A Logic File is a Tab delimited text file. Any lines not recognized as a valid Key/Value pair, will be ignore and can be used as remarks/other.

The Logic File uses the syntax.

Syntax: **[Key]** [tab] **[Value]** ... [tab] **[Value]**

Example: **lcc:key** **value**

Any extra tabs in a line after the expected ones are considered remarks and will be ignored. This is a nice way to document specific Key settings (see Log Levels in the Logic File example(s) for reference). Also, if you place a tab before a line, that will essentially make it a remark and will be ignored, which makes using/not using logic without removing quicker.,

Any line not starting with an expected key is ignored, which makes placing remarks/formatting the logic easy.

20191024: with the new User Tools module, user Uploads or Invites are now stored in their own Logic Files.

Logic File

There are multiple keys that have a recommendation to encrypt their values. To encrypt those values, use the program **lccEncryptLogicValues**, ref:

<http://services4.lowercolumbia.edu/demo/projectlid/LCCEncryptLogicValues>

Just supply the same Hash, Salt, and Key values you supply in the Logic File below.

Any value supplied to a key can be encrypted if you supply the following keys:

- **lcc:sourceEncryptedPasswordHash**
- **lcc:sourceEncryptedSaltKey**
- **lcc:sourceEncryptedVIKey**

If encryption used, prepend any value with "[lcc:encrypted]", plus the encrypted value.

Example: **lcc:user** **[lcc:encrypted]EGFDEFGFEFGDDEG**

lcc:debugIP (optional)(none to many per Logic File)

If access the URL and your IP matches this key value, extra information may be displayed. Use the 'lcc:debugLevel' key to control what information is displayed to the Debug IP(s).

Syntax: `lcc:debugIP [tab] [#.#.#.#]`

Example: `lcc:debugIP 123.123.123.123`

lcc:debugLevel *(optional)(none to many per Logic File)*

Controls what the displays to the Debug IP(s).

Debug Levels

- 1 Show Logic Name/Values: what names/values are read from the Logic File
- 2 Show Query Names/Values: what names/values are supplied in the URL
- 3 Show post name/values
- 4 Show environment variables
- 5 Show user upload/download
- 7 Show process post content
- 8 Show LDAP
- 9 Show Process Mode
- 10 Show Validate Portal LDAP Key
- 11 Show Parse Encrypted
- 12 Show Invitations
- 13 Show Logic Paths loading
- 14 Show Email
- 15 Show Upload Expirations
- 16 Show Email Templates

Syntax: `lcc:debugLevel [tab] [#]`

Example: `lcc:debugLevel 2`

lcc:logPath *(optional)(none to many per Logic File)*

If provided, will log information about program usage to this location. Make sure to open up file rights to the App Pool user. The program will automatically append the Year/Month/Date and ".log" to the end of the path, i.e. "-20160830.log".

Syntax: `lcc:logPath [tab] [Path]`

Example: `lcc:logPath F:\folder\lccRetrievedSecuredFileLog`

lcc:programURL *(mandatory)(one per Logic File)*

The relative URL of the program on the website.

Syntax: `lcc:programURL [tab] [URL]`

Example: `lcc:programURL /scripts/lccRetrieveSecuredFile.exe`

lcc:maxDownloadsPath *(optional)(one per Set)*

Provides a path to keep stats on Sets that have defined the key 'lcc:securedFileSetMaxDownloads'. The program will keep two sets of files for each Set that has download attempts.

File Sets

- `attempted-[Set Id].txt`: appends an 'X' for each attempt at downloading. The size of the file tells the program how many attempts.
- `downloaded-[Set Id].txt`: appends an 'X' for each approved download. The size of the file tells the program how many attempts.

Syntax: `lcc:maxDownloadsPath [tab] [Path]`

Example: `lcc:maxDownloadsPath 2`

lcc:securityKey *(optional)(one per Logic File)*

Must be provided if using the User Tools module.

Provides the key used to encrypt/decrypt any encrypted values in the Logic File. See the section Logic File Encrypted Values.

Syntax: `lcc:securityKey [tab] [Key]`

Example: `lcc:securityKey our super secret key`

lcc:LDAPServer *(optional)(one to many per Logic File)*

Must be provided if using the User Tools module.

Provides LDAP servers in standard LDAP format. If more than one provided, will try each one until one works.

LDAP standard format: `[server].[domain]/dc=[domani],dc=[domain]`

Syntax: `lcc:LDAPServer [tab] [LDAP path]`

Example: `lcc:LDAPServer ourServer.college.edu/dc=college,dc=edu`

lcc:LDAPUserId *(optional)(one per Logic File)*

Must be provided if using the User Tools module.

Provides the account id to perform LDAP actions. This account only needs read-only to the domain. Though the logging in user's credentials are used initially, all pages after log-in use this account to look up information on the user, like DisplayName, SAMAccountName, etc.

We recommend encrypting this value.

Syntax: **lcc:LDAPUserId** *[tab]* **[User Id]**

Example: **lcc:LDAPUserId** **jdoe**

Example (encrypted):

lcc:LDAPUserId **[lcc:encrypted]ASDFASDF**

lcc:LDAPUserPassword *(optional)(one per Logic File)*

Must be provided if using the User Tools module.

Provides the account password to perform LDAP actions.

We recommend encrypting this value.

Syntax: **lcc:LDAPUserPassword** *[tab]* **[Password]**

Example: **lcc:LDAPUserPassword** **jdoe**

Example (encrypted):

lcc:LDAPUserPassword **[lcc:encrypted]ASDFASDF**

lcc:LDAPSecurityGroupUsers *(optional)(one to many per Logic File)*

Must be provided if using the User Tools module.

Provides what security group(s) are allowed to use the program.

The value is in Distinguished Name (DN) format.

Syntax: **lcc:LDAPSecurityGroupUsers** *[tab]* **[Group]**

Example: **lcc:LDAPSecurityGroupUsers** **CN=ourGroup,DC=college,DC=edu**

Example (multiple):

lcc:LDAPSecurityGroupUsers CN=ourGroup,DC=college,DC=edu
lcc:LDAPSecurityGroupUsers CN=ourGroup2,DC=college,DC=edu

lcc:SMTP (optional, one to many per Logic File)

Must be provided if using the User Tools module.

Defines settings for SMTP (email). These are used to email both the recipient and send about uploads and invites.

Provide one of these keys for each desired SMTP setting.

Note: the SMTP settings for 'from' and 'to' will be dynamically configured by the program depending on the upload, invite submissions.

Settings Available

- **Server:** the email server name/ip.
ex: ouremailserver.edu
- **Port:** the email server port.
ex: 25
- **SSL:** (optional) if your email server supports SSL connection. The only valid value is 'YES'.
- **MailboxId:** (optional) the mailbox id to log into the email server. If not provided, the connection is assumed to be 'no authentication required', like IP based.
- **MailboxPassword:** (optional) the mailbox password.
- **Subject:** the subject to show on the message. This will show at the beginning of the subject for all emails, while the program will append different subject content depending on the actions.
- **MonitorName:** if you want to monitor the emails being sent to the recipient and sender, provide the name of the person who should receive monitoring emails. You can provide one to many of these. This must be followed by the key "MonitorEmail".
- **MonitorEmail:** if you want to monitor the emails being sent to the recipient and sender, provide the email of the person who should receive monitoring emails. You can provide one to many of these. This must be preceded by the key "MonitorName".
- **EmailTemplate:** specifies what email templates to use. All templates must be defined. This key has four (4) columns:
Syntax: lcc:SMTP [tab] [..Key..] [tab] [..Sub-Key..] [tab] [..Value..]

The Sub-Keys available are:

- UploadFileToRecipientTemplatePath : used when emailing the recipient about an uploaded file.
- UploadFileToSenderTemplatePath : used when emailing the sender about an uploaded file.
- UploadMessageToRecipientTemplatePath : used when emailing the recipient about an uploaded message.
- UploadMessageToSenderTemplatePath : used when emailing the sender about an uploaded message.
- UploadInviteToRecipientTemplatePath : used when emailing the recipient about an invite.
- UploadInviteToSenderTemplatePath : used when emailing the sender about an invite.

Example: lcc:SMTP EmailTemplate UploadFileToRecipientTemplatePath E:\folder\template.txt

Syntax: lcc:SMTP [tab] [..Key..] [tab] [..Value..]

Example: lcc:SMTP Server ouremailserver.edu

Example #2 (multiple):

```
lcc:SMTP      Server ouremailserver.edu
lcc:SMTP      Port      25
              lcc:SMTP      SSL      YES
              lcc:SMTP      MailboxId      ...
              lcc:SMTP      MailboxPassword      ...
lcc:SMTP      Subject      lccDiskWatcher Alert: Monitor Executables
```

Example #3 (multiple, with a SMTP flag):

```
lcc:SMTP      Server ouremailserver.edu
lcc:SMTP      Port      25
lcc:SMTP      Subject      lccDiskWatcher Alert: Monitor Executables
lcc:SMTP      Flag      Do Not Send File Names      YES
```

Example #3 (multiple, with a SMTP flag):

```
lcc:SMTP      Server ouremailserver.edu
lcc:SMTP      Port      25
lcc:SMTP      Subject      lccDiskWatcher Alert: Monitor Executables
lcc:SMTP      EmailTemplate UploadFileToRecipientTemplatePath    E:\folder\template.txt
```

lcc:includePath (optional)(one to many per Logic File)

Content you want included in the HEAD section of your page.

Syntax: lcc:includePath [tab] [Path]

Example: lcc:includePath e:\folder\data\file.htm

lcc:topperPath (optional)(one to many per Logic File)

Content you want included in the top of the BODY section of your page.

Syntax: `lcc:topperPath [tab] [Path]`

Example: `lcc:topperPath e:\folder\data\topper.htm`

lcc:footerPath *(optional)(one to many per Logic File)*

Content you want included in the bottom of the BODY section of your page.

Syntax: `lcc:footerPath [tab] [Path]`

Example: `lcc:footerPath e:\folder\data\footer.htm`

lcc:securedFilePath *(optional)(one per Logic File)*

Must be provided if using the User Tools module.

The default location for where to place/look for secured files. This is used if not defined in a Secured File Set, see Key `lcc:securedFilePath`.

Syntax: `lcc:securedFilePath [tab] [Path]`

Example: `lcc:securedFilePath e:\folder\securedFiles`

lcc:uploadLogicsPath *(optional)(one per Logic File)*

Must be provided if using the User Tools module.

Where User Tools individual Logic Files will be stored.

Syntax: `lcc:uploadLogicsPath [tab] [Path]`

Example: `lcc:uploadLogicsPath e:\folder\lccRetrieveSecuredFile-logics`

lcc:securedFilesDeniedPath *(optional)(one per Logic File)*

Must be provided if using the User Tools module.

What file will be displayed if the 'key' is invalid. If the 'id' or URL is incorrect, an onscreen message will alert the End User. If the 'key' is invalid, this file will be returned allowing you to construct the 'denied' message you want.

Syntax: `lcc:securedFilesDeniedPath [tab] [Path]`

Example: `lcc:securedFilesDeniedPath F:\folder\messages\deniedAccessToSecuredContent.htm`

lcc:userUploadDefaultEmailDomain *(optional)(one per Logic File)*

Must be provided if using the User Tools module.

What email domain will be appended to a logged in user's SAMAccountName to build the From email address.

Syntax: **lcc:userUploadDefaultEmailDomain** *[tab]* **[Domain]**

Example: **lcc:userUploadDefaultEmailDomain** **@ourcollege.edu**

lcc:userUploadAllowManualMaxDownloads *(optional)(one per Logic File)*

Only used with the User Tools module.

If not provided, the Max Downloads is hard set to '1'. If provided, the user uploading the file can choose different amounts. Invitees are still hard set to '1'.

Valid Values

- YES

Syntax: **lcc:userUploadAllowManualMaxDownloads** *[tab]* **[YES]**

Example: **lcc:userUploadAllowManualMaxDownloads** **YES**

lcc:userUploadMaxSize *(optional)(one per Logic File)*

Only used with the User Tools module.

Sets the maximum size a file can be in uploading, in bytes. Ex: 5000000 would be 5 MB.

If not provided, the Max Size of upload is '1000000' (i.e. 1 MB).

Syntax: **lcc:userUploadMaxSize** *[tab]* **[#]**

Example: **lcc:userUploadMaxSize** **5000000**

lcc:userUploadExpirationDays *(optional)(one per Logic File)*

Only used with the User Tools module.

Sets the maximum amount of days an Upload is valid for. If not used after those days, it will be invalidated.

If not provided, the Max Days is set to '30'.

Syntax: `lcc:userUploadExpirationDays [tab] [#]`

Example: `lcc:userUploadExpirationDays 30`

lcc:userInvitationExpirationDays *(optional)(one per Logic File)*

Only used with the User Tools module.

Sets the maximum amount of days an Invitation is valid for. If not used after those days, it will be invalidated.

If not provided, the Max Days is set to '7'.

Syntax: `lcc:userInvitationExpirationDays [tab] [#]`

Example: `lcc:userInvitationExpirationDays 7`

lcc:userResponseUploadFile *(optional)(one per Logic File)*

Only used with the User Tools module.

Changes the response message on the web page after a user has uploaded a file.

Syntax: `lcc:userResponseUploadFile [tab] [...]`

Example: `lcc:userResponseUploadFile The file and link to download has been successfully sent to the recipient.`

lcc:userResponseUploadMessage *(optional)(one per Logic File)*

Only used with the User Tools module.

Changes the response message on the web page after a user has uploaded a message.

Syntax: `lcc:userResponseUploadMessage [tab] [...]`

Example: `lcc:userResponseUploadMessage The message and link to view has been successfully sent to the recipient.`

lcc:userResponseUploadInvitation *(optional)(one per Logic File)*

Only used with the User Tools module.

Changes the response message on the web page after a user has uploaded an invitation.

Syntax: `lcc:userResponseUploadInvitation [tab] [...]`

Example: `lcc:userResponseUploadFile` The invitation to upload a file has been successfully sent to the recipient.

lcc:userUploadMimeTypeAuthorized (optional)(one per Logic File)

Must be provided if using the User Tools module.

Sets what MIME Types are allowed to be uploaded. One to many can be provided. See distribution site for common MIME types.

Syntax: `lcc:userUploadMimeTypeAuthorized [tab] [File Extension] [tab] [MIME Type]`

Example: `lcc:userUploadMimeTypeAuthorized bmp image/bmp`

Example (multiple):

`lcc:userUploadMimeTypeAuthorized bmp image/bmp`

`lcc:userUploadMimeTypeAuthorized pdf application/pdf`

lcc:securedFileSetId (mandatory)(one to many per Logic File)

Starts a new set (file) of settings. This must be the first key for each set/file being supplied. All "lcc:securedFileSet..." keys supplied after this, until the next "lcc:securedFileSetId" are settings for this set/file.

The value can be anything, it is used to report what set it is working on, and what file the Web Viewer's URL specifies they wish to download.

Syntax: `lcc:securedFileSetId [tab] [...]`

Example: `lcc:securedFileSetId ourSecuredFile1`

lcc:securedFileSetShowFilename (mandatory)(one per Set)

What filename will be provided to the end user browser. This allows you to control what the filename is on the file being downloaded. It can be the same as the actual file, or anything you make up.

Syntax: `lcc:securedFileSetShowFilename [tab] [...]`

Example: `lcc:securedFileSetShowFilename ASecuredFileForYOU.txt`

lcc:securedFileSetContentType *(mandatory)(one per Set)*

What Content Type will be reported back to the End User browser for this file.

Media Types are defined by the IANA.org at:

<https://www.iana.org/assignments/media-types/media-types.xhtml>

Some common ones are:

- image/jpeg
- text/plain
- text/html
- application/pdf

Syntax: **lcc:securedFileSetContentType** *[tab]* [...]

Example: **lcc:securedFileSetContentType** text/plain

lcc:securedFileSetPath *(mandatory)(one per Set)*

What path is used to retrieve the file.

Syntax: **lcc:securedFileSetPath** *[tab]* [Path]

Example: **lcc:securedFileSetPath** F:\folder\data\securedFile.txt

Example #2: **lcc:securedFileSetPath** \\server\share\data\securedWebPage.htm

lcc:securedFileSetKey *(mandatory)(one per Set)*

What 'key' will be needed by the End User viewer (URL) to access this file.

Syntax: **lcc:securedFileSetKey** *[tab]* [...]

Example: **lcc:securedFileSetKey** ourSecretKey

lcc:securedFileSetDeniedPath *(mandatory)(one per Set)*

What file will be displayed if the 'key' is invalid. If the 'id' or URL is incorrect, an onscreen message will alert the End User. If the 'key' is invalid, this file will be returned allowing you to construct the 'denied' message you want. You can use the same on multiple Sets if desired.

Syntax: **lcc:securedFileSetDeniedPath** *[tab]* [Path]

Example: `lcc:securedFileSetDeniedPath` `F:\folder\messages\deniedAccessToSecuredContent.htm`

lcc:securedFileSetValidOnOrAfter *(optional)(one per Set)*

If provided, access to the file will only be allowed On or After this date. If not in a valid date range, the set 'id' will not be valid.

Must be in format: **M/D/YYYY**

Syntax: `lcc:securedFileSetValidOnOrAfter [tab] [M/D/YYYY]`

Example: `lcc:securedFileSetValidOnOrAfter 8/30/2016`

lcc:securedFileSetValidOnOrBefore *(optional)(one per Set)*

If provided, access to the file will only be allowed On or Before this date. If not in a valid date range, the set 'id' will not be valid.

Must be in format: **M/D/YYYY**

Syntax: `lcc:securedFileSetValidOnOrBefore [tab] [M/D/YYYY]`

Example: `lcc:securedFileSetValidOnOrBefore 8/30/2016`

lcc:securedFileSetAttachmentModeOff *(optional)(one per Set)*

If provided and has a valid value, will turn off Attachment Mode. Use this if you want the file displayed directly, i.e. in a web browser. This comes in handy if you want to return a web page or other browser friendly content.

Valid Values

YES

Syntax: `lcc:securedFileSetAttachmentModeOff [tab] [YES]`

Example: `lcc:securedFileSetAttachmentModeOff YES`

lcc:securedFileSetEncrypted *(optional)(one per Set)*

If provided and has a valid value, will return an encrypted file. When retrieving the file for the end user, the file will be decrypted as it is provided. The file contents are never stored in an unencrypted state.

To provide an encrypted file, please see the section 'Encrypting A File'.

This key requires that the following keys also be provided and match the values supplied when encrypting the file:

- `lcc:sourceEncryptedPasswordHash`
- `lcc:sourceEncryptedSaltKey`
- `lcc:sourceEncryptedVIKey`

Valid Values

YES

Syntax: `lcc:securedFileSetEncrypted [tab] [YES]`

Example: `lcc:securedFileSetEncrypted YES`

`lcc:securedFileSetMaxDownloads` (optional)(one per Set)

If provided and has a number ≥ 1 , the program will keep stats on how many times the file has been retrieved. If that stat exceeds this key's number value, the download will be aborted and the End User will receive a message saying the Maximum downloads have been reached. Must supply the key 'lcc:maxDownloadsPath' to use this key.

Syntax: `lcc:securedFileSetMaxDownloads [tab] [#]`

Example: `lcc:securedFileSetMaxDownloads 2`

`lcc:securedFileSetMaxDownloadsPath` (optional)(one per Set)

Provides a path to keep stats on Sets that have defined the key 'lcc:securedFileSetMaxDownloads'. The program will keep two sets of files for each Set that has download attempts.

File Sets

- `attempted-[Set Id].txt`: appends an 'X' for each attempt at downloading. The size of the file tells the program how many attempts.
- `downloaded-[Set Id].txt`: appends an 'X' for each approved download. The size of the file tells the program how many attempts.

Syntax: `lcc:securedFileSetMaxDownloadsPath [tab] [Path]`

Example: `lcc:securedFileSetMaxDownloadsPath 2`

`lcc:sourceEncryptedPasswordHash` (optional)(one per Logic File)

Hash value supplied for encryption. Must be used with keys 'lcc:sourceEncryptedSaltKey' and 'lcc:sourceEncryptedVIKey'.

Any set of characters can be used.

Syntax: `lcc: sourceEncryptedPasswordHash [tab] [...]`

Example: `lcc: sourceEncryptedPasswordHash s0m3C00lPhr@$e`

lcc:sourceEncryptedSaltKey (optional)(one per Logic File)

Salt Key value supplied for encryption. Must be used with keys 'lcc:sourceEncryptedPasswordHash' and 'lcc:sourceEncryptedVIKey'.

Any set of characters can be used.

Syntax: `lcc: sourceEncryptedSaltKey [tab] [...]`

Example: `lcc: sourceEncryptedSaltKey s0m3C00lPhr@$e`

lcc:sourceEncryptedVIKey (optional)(one per Logic File)

VI Key value supplied for encryption. Must be used with keys 'lcc:sourceEncryptedPasswordHash' and 'lcc:sourceEncryptedSaltKey'.

Any set of characters can be used, but, must be exactly 16 characters long.

Syntax: `lcc:sourceEncryptedVIKey [tab] [...]`

Example: `lcc:sourceEncryptedVIKey @$e12`

Encrypt A File

To encrypt a file use the key 'lcc:encryptFilePath' in the command line. In this mode, it will encrypt the file and return back to the command window.

In this mode, the program will only encrypt the file and perform no other functions.

For the program to encrypt the file, you must supply the keys (see above for details on their syntax/example) in a Logic File:

- lcc:sourceEncryptedPasswordHash
- lcc:sourceEncryptedSaltKey
- lcc:sourceEncryptedVIKey

These same keys must be provided with the same values in the Logic File on your web server.

Normally, you would have a Logic File specific for Encrypting a file, and another used on your web server for end user processing.

Syntax: `lccRetrieveSecuredFile.exe lcc:encryptFilePath [...]`

Example: `lccRetrieveSecuredFile.exe lcc:encryptFilePath "ourSecretFile.zip"`

Logic File Encrypted Values

Logic File Examples

Basic Retrieve Examples

```
lcc:debugIP 123.123.123.123
lcc:debugLevel 1 logic name/values
lcc:debugLevel 2 query name/values

lcc:logPath f:\folder\logs\lccRetrieveSecuredFileLogs\lccRetrieveSecuredFile

----- this set will download a text file -----
lcc:securedFileSetId ourSecretFile1
lcc:securedFileSetShowFilename aSecretFileForYOU.txt
lcc:securedFileSetContentType text/plain
lcc:securedFileSetPath f:\folder\data\securedFile.txt
lcc:securedFileSetKey ourSecretKey
lcc:securedFileSetDeniedPath f:\folder\messages\deniedAccess.txt

----- this set will download a web page, but, only from 8/25/2016 to 9/5/2016 -----
lcc:securedFileSetId ourSecretFile2
lcc:securedFileSetShowFilename aSecretFileForYOU.htm
lcc:securedFileSetContentType text/html
lcc:securedFileSetPath f:\folder\data\securedFile.htm
lcc:securedFileSetKey ourSecretKey
```

```
lcc:securedFileSetDeniedPath    f:\folder\messages\deniedAccess.txt
lcc:securedFileSetValidOnOrAfter 8/25/2016
lcc:securedFileSetValidOnOrBefore 9/5/2016

----- this set will display a web page, but, only after 8/25/2017 -----
lcc:securedFileSetId           ourSecretFile2
lcc:securedFileSetShowFilename  aSecretFileForYOU.htm
lcc:securedFileSetContentType   text/html
lcc:securedFileSetPath         f:\folder\data\securedFile.htm
lcc:securedFileSetKey          ourSecretKey
lcc:securedFileSetDeniedPath    f:\folder\messages\deniedAccess.txt
lcc:securedFileSetValidOnOrAfter 8/25/2017
lcc:securedFileSetAttachmentModeOff  YES

----- this set will display a web page, but, only 2 times -----
lcc:securedFileSetId           ourSecretFile2
lcc:securedFileSetShowFilename  aSecretFileForYOU.htm
lcc:securedFileSetContentType   text/html
lcc:securedFileSetPath         f:\folder\data\securedFile.htm
lcc:securedFileSetKey          ourSecretKey
lcc:securedFileSetDeniedPath    f:\folder\messages\deniedAccess.txt
lcc:securedFileSetMaxDownloads  2
lcc:securedFileSetAttachmentModeOff  YES
```

Encrypted File Retrieve Example

```
lcc:debugIP 123.123.123.123
lcc:debugLevel 1 logic name/values
lcc:debugLevel 2 query name/values

lcc:logPath f:\folder\logs\lccRetrieveSecuredFileLogs\lccRetrieveSecuredFile

lcc:sourceEncryptedPasswordHash s0m3C001Phr@$e
lcc:sourceEncryptedSaltKey s0m3C001Phr@$e
lcc:sourceEncryptedVIKey s0m3C001Phr@$e12

----- this set will download a text file -----
lcc:securedFileSetId           ourSecretFile1
lcc:securedFileSetShowFilename  aSecretFileForYOU.txt
lcc:securedFileSetContentType   text/plain
lcc:securedFileSetPath         f:\folder\data\securedFile.txt
lcc:securedFileSetKey          ourSecretKey
lcc:securedFileSetEncrypted     YES
```

lcc:securedFileSetDeniedPath f:\folder\messages\deniedAccess.txt

Enable User Tools

```
lcc:debugIP 123.123.123.123
  lcc:debugLevel 1 logic name/values
  lcc:debugLevel 2 query name/values
  lcc:debugLevel 3 post name/values
  lcc:debugLevel 4 environment variables
  lcc:debugLevel 5 user upload/download
  lcc:debugLevel 7 process post content
  lcc:debugLevel 8 LDAP
  lcc:debugLevel 9 Process Mode
  lcc:debugLevel 10 Validate Portal LDAP Key
  lcc:debugLevel 11 Parse Encrypted
  lcc:debugLevel 12 Invitations
  lcc:debugLevel 13 Logic Paths loading
  lcc:debugLevel 14 Email

lcc:logPath [drive]:\folder\lccRetrieveSecuredFileLogs\lccRetrieveSecuredFile
lcc:maxDownloadsPath [drive]:\web\logs\lccRetrieveSecuredFileLogs\maxDownloads

lcc:sourceEncryptedPasswordHash ...
lcc:sourceEncryptedSaltKey ...
lcc:sourceEncryptedVIKey ...

lcc:LDAPServer ourdomain.college.edu/dc=college,dc=edu

lcc:securityKey our super secret key
lcc:LDAPUserId [lcc:encrypted]...
lcc:LDAPUserPassword [lcc:encrypted]...

lcc:LDAPSecurityGroupUsers CN=lccRetrieveSecuredFile-users,OU=Security Groups,DC=ourCollege,DC=edu
lcc:LDAPSecurityGroupUsers CN=allStaff,OU=Security Groups,DC=ourCollege,DC=edu

lcc:SMTP Server ourEmailServer.ourCollege.edu
lcc:SMTP Port 25
  lcc:SMTP SSL YES
  lcc:SMTP MailboxId ...
  lcc:SMTP MailboxPassword ...

lcc:SMTP Subject LCC Secured Content:
```

```
lcc:SMTP      EmailTemplate UploadFileToRecipientTemplatePath e:\lccRSF-files\uploadFileToRecipient.htm
lcc:SMTP      EmailTemplate UploadFileToSenderTemplatePath   e:\lccRSF-files\uploadFileToSender.htm
lcc:SMTP      EmailTemplate UploadMessageToRecipientTemplatePath e:\lccRSF-files\uploadMessageToRecipient.htm
lcc:SMTP      EmailTemplate UploadMessageToSenderTemplatePath e:\lccRSF-files\uploadMessageToSender.htm
lcc:SMTP      EmailTemplate UploadInviteToRecipientTemplatePath e:\lccRSF-files\uploadInviteToRecipient.htm
lcc:SMTP      EmailTemplate UploadInviteToSenderTemplatePath   e:\lccRSF-files\uploadInviteToSender.htm
```

```
lcc:SMTP      MonitorName    lccRSF Monitor #2 - John Doe
lcc:SMTP      MonitorEmail  jdoe@ourcollege.edu
lcc:SMTP      MonitorName    lccRSF Monitoring
lcc:SMTP      MonitorEmail  lccRSFMonitoring@ourcollege.edu
```

```
lcc:programURL /scripts/lccRetrieveSecuredFile.exe
lcc:includePath [drive]:\web\pages\virtual\header-1.htm
lcc:topperPath  [drive]:\web\pages\lccRetrieveSecuredFile\topper.htm
lcc:footerPath [drive]:\web\pages\lccRetrieveSecuredFile\footer.htm
```

```
lcc:securedFilesPath [drive]:\data\lccRetrieveSecuredFile-files\userUploads
lcc:uploadLogicsPath [drive]:\data\lccRetrieveSecuredFile-logics
lcc:securedFilesDeniedPath [drive]:\web\pages\lccRetrieveSecuredFile\securedFileDenied-general.txt
lcc:userUploadDefaultEmailDomain @ourcollege.edu
lcc:userUploadAllowManualMaxDownloads YES
lcc:userUploadMaxSize 6000000
lcc:userInvitationExpirationDays 7
```

```
lcc:userUploadMimeTypeAuthorized bmp      image/bmp
lcc:userUploadMimeTypeAuthorized csv      text/csv
lcc:userUploadMimeTypeAuthorized doc      application/msword
lcc:userUploadMimeTypeAuthorized docm    application/vnd.ms-word.document.macroenabled.12
lcc:userUploadMimeTypeAuthorized docx    application/vnd.openxmlformats-officedocument.wordprocessingml.document
lcc:userUploadMimeTypeAuthorized dotm    application/vnd.ms-word.template.macroenabled.12
lcc:userUploadMimeTypeAuthorized dotx    application/vnd.openxmlformats-officedocument.wordprocessingml.template
lcc:userUploadMimeTypeAuthorized eml     message/rfc822
lcc:userUploadMimeTypeAuthorized jpeg    image/jpeg
lcc:userUploadMimeTypeAuthorized jpg     image/jpeg
lcc:userUploadMimeTypeAuthorized htm     text/html
lcc:userUploadMimeTypeAuthorized html    text/html
lcc:userUploadMimeTypeAuthorized mp4     video/mp4
lcc:userUploadMimeTypeAuthorized mpp     application/vnd.ms-project
lcc:userUploadMimeTypeAuthorized pdf     application/pdf
lcc:userUploadMimeTypeAuthorized potm    application/vnd.ms-powerpoint.template.macroenabled.12
lcc:userUploadMimeTypeAuthorized ppsm    application/vnd.ms-powerpoint.slideshow.macroenabled.12
lcc:userUploadMimeTypeAuthorized pptm    application/vnd.ms-powerpoint.presentation.macroenabled.12
lcc:userUploadMimeTypeAuthorized pptx    application/vnd.openxmlformats-officedocument.presentationml.presentation
```

```
lcc:userUploadMimeTypeAuthorized pub application/x-mspublisher
lcc:userUploadMimeTypeAuthorized pdf application/pdf
lcc:userUploadMimeTypeAuthorized png image/png
lcc:userUploadMimeTypeAuthorized rtf application/rtf
lcc:userUploadMimeTypeAuthorized svg image/svg+xml
lcc:userUploadMimeTypeAuthorized tar application/x-tar
lcc:userUploadMimeTypeAuthorized tiff image/tiff
lcc:userUploadMimeTypeAuthorized txt text/plain
lcc:userUploadMimeTypeAuthorized wav audio/x-wav
lcc:userUploadMimeTypeAuthorized xls application/vnd.ms-excel
lcc:userUploadMimeTypeAuthorized xlsm application/vnd.ms-excel.sheet.macroenabled.12
lcc:userUploadMimeTypeAuthorized xlsx application/vnd.openxmlformats-officedocument.spreadsheetml.sheet
lcc:userUploadMimeTypeAuthorized xml application/xml
lcc:userUploadMimeTypeAuthorized zip application/zip
```

Definitions

CSS - Cascading Style Sheets

DN - Distinguished Name

HTML - Hyper-Text Markup Language

IIS - Internet Information Servers (Microsoft web server)

LDAP - Lightweight Directory Access Protocol

MIME - Multipurpose Internet Mail Extensions

SMTP - Simple Mail Transfer Protocol

Modifications

NAME	DATE	MODIFICATION
David Mielcarek	8/30/2016	Created
David Mielcarek	3/20/2017	Added keys 'lcc:sourceEncryptedPasswordHash', 'lcc:sourceEncryptedSaltKey', 'lcc:sourceEncryptedVIKey', 'lcc:securedFileSetEncrypted'
David Mielcarek	20191015	Added User module, many new keys added to support
David Mielcarek	20191028	Added key 'lcc:userUploadExpirationDays'

David Mielcarek	20191030	Added email templates, see key lcc:SMTP
David Mielcarek	20191104	Added keys lcc:userResponseUpload(File/Message/Invitation)

End of document