

**Check Image Parsing Script**

**Reference Guide**

Table of Contents:

[Name: 4](#_Toc346188612)

[Description: 4](#_Toc346188613)

[Synopsis: 4](#_Toc346188614)

[Example: 4](#_Toc346188615)

[Control Files: 4](#_Toc346188616)

[Configuration Options: 4](#_Toc346188617)

[Configuration Options Notes: 5](#_Toc346188618)

[Configuration Options List: 5](#_Toc346188619)

[data\_file\_format 5](#_Toc346188620)

[dir\_infoftp 5](#_Toc346188621)

[dir\_ftpbu 6](#_Toc346188622)

[dir\_process 6](#_Toc346188623)

[dir\_log 6](#_Toc346188624)

[dir\_backup 6](#_Toc346188625)

[dir\_image 6](#_Toc346188626)

[reg\_ex\_xfer\_fname 7](#_Toc346188627)

[reg\_ex\_img\_fname 7](#_Toc346188628)

[reg\_ex\_encrypt\_ext 7](#_Toc346188629)

[reg\_ex\_zip\_ext 7](#_Toc346188630)

[reg\_ex\_img\_ext 7](#_Toc346188631)

[reg\_ex\_idx\_ext 8](#_Toc346188632)

[reg\_ex\_file\_prefix 8](#_Toc346188633)

[xfer\_file\_cnt 8](#_Toc346188634)

[img\_file\_cnt 8](#_Toc346188635)

[request\_download 9](#_Toc346188636)

[request\_xfer\_chk 9](#_Toc346188637)

[request\_decrypt 9](#_Toc346188638)

[request\_unzip 9](#_Toc346188639)

[request\_chksum 9](#_Toc346188640)

[request\_parse\_log 10](#_Toc346188641)

[request\_img\_subdir 10](#_Toc346188642)

[pwd\_unzip 10](#_Toc346188643)

[ftp\_host 10](#_Toc346188644)

[ftp\_usr 11](#_Toc346188645)

[ftp\_pwd 11](#_Toc346188646)

[ftp\_path 11](#_Toc346188647)

[xfer\_time\_interval 11](#_Toc346188648)

[maillist\_tag\_int 11](#_Toc346188649)

[maillist\_tag\_ext 12](#_Toc346188650)

[img\_fname\_format 12](#_Toc346188651)

[err\_flag\_file 12](#_Toc346188652)

[maillist\_file 13](#_Toc346188653)

[chksum\_log\_file 13](#_Toc346188654)

[parsing\_log\_file 13](#_Toc346188655)

[lock\_file 13](#_Toc346188656)

[Control File Example: 13](#_Toc346188657)

[Process Flow: 14](#_Toc346188658)

[Process Logs: 14](#_Toc346188659)

[Library Files: 15](#_Toc346188660)

# Name:

**get\_chk\_image.pl** – Check image files parsing script.

# Description:

This script parses data files containing multiple check images and creates a separate output file for each parsed check image. Multiple image data file formats are supported, which are described later in this document. The image files are always created in TIFF format.

# Synopsis:

**get\_chk\_image.pl -c <*cid>*  [ -f <*control\_file>* ] [ --debug ] [ --help ]**

Command line options:

**-c** <cid>

Defines the process *CID*. (Required)

**-f** <control\_file>

Full pathname of control file used to override default process flow.

**--debug**

Enables debug mode.

**--help**

Prints a script usage message.

# Example:

/home/master/get\_chk\_image.pl -c svbt -f svbt\_image.control 2>&1 | tee –a /z/prep/svbt/log/get\_chk\_image.log

# Control Files:

In order to make the script flexible and adaptable to different process requirements, a set of configuration options can be set to control the script’s process flow. These configuration options have default values already defined in the script, but can be reset in a separate control file that can be referenced by the script.

By default, the script looks for a control file named **/home/control/{CID}\_image.control** (*Ex. /home/control/svbt\_image.control*). The location of the control file can be redefined using the **–f** command line option. If the control file is not specified and the default control file doesn’t exist, then the script uses the default values defined in the script.

# Configuration Options:

The configuration options are implemented as a set of key/value pairs both in the script and in the optional control file. In the script, they are predefined as *key/value pairs* of the ***%CFG*** hash (*Ex. data\_file\_format => ‘ISR’*). In the control file, they are defined in the format of *key=value records* (*Ex. data\_file\_format = ISR*).

# Configuration Options Notes:

* Only valid keys are accepted or an exception is generated.
* Quoting string values is not required. Ex. *data\_file\_format = ISR* is sufficient.
* If the value is missing, as in *data\_file\_format = “”*, the default value will be used.
* Any text following the pound sign ‘#’ is treated as comments.
* Some options are used to define path names. For example the options prefixed by “**dir\_**” (*Ex. dir\_ftpbu*). The value assigned to them is interpreted as being relative to a default path. If a reference to an environment variable, as in ***$ENV{var\_name}***, is contained anywhere within the value, the value is interpreted as being an absolute path name and the referenced environment variable is evaluated accordingly. For example, the string *$ENV{home\_keep}/$ENV{keep\_dir}/test* will be evaluated directly as */home/keep/test* and not as */d/home/keep/test* if the default directory for that option is *$ENV{d\_dir}.*

# Configuration Options List:

Name:

### data\_file\_format

Description:

Image data file format. Valid values are:*ISR, X937, COFF,* and *JHA.*

**ISR**: ISRetrieve. This format uses a text index file and binary image file pair.

Currently used by the following clients: FNBS, SJCB, SVBT, CAMB.

**X937:** X9.37 Standard Image Cash Letter. This format uses a single ebcdic binary data file.

Currently used by the following clients: PROV, BKFN, NRBK.

**COFF:** Common Object File Format (COFF). This format uses a single variable length record file.

Currently used by the following clients: TKCU.

**JHA:** Jack Henry and Associates. This format uses a index xml file and multiple binary image files set.

Currently used by the following clients: MSBK, BANB, ISBK.

Default:

**data\_file\_format = ISR**

Name:

### dir\_infoftp

Description:

Directory where the customer transfers the data files.

Default:

**dir\_infoftp = $ENV{d\_dir}/infoftps2/${CID}**

Name:

### dir\_ftpbu

Description:

Customer data file pickup directory where the files are placed by the file scanner.

Default:

**dir\_ftp = $ENV{d\_dir}/ftpbu/${CID}bu**

Name:

### dir\_process

Description:

Data file processing directory. Any files that exist in this directory will be removed by the script before processing.

Default:

**dir\_process = $ENV{d\_dir}/${CID}/processing**

Name:

### dir\_log

Description:

Directory where to create the process log files.

Default:

**dir\_log = $ENV{d\_dir}/prep/${CID}/log**

Name:

### dir\_backup

Description:

Directory where to archive the processed data files.

Default:

**dir\_backup = $ENV{d\_dir}/download/${CID}/image\_files**

Name:

### dir\_image

Description:

Directory where TIFF check image files are to be created.

Default:

**dir\_image = $ENV{d\_dir}/prep/${CID}/tiff**

Name:

### reg\_ex\_xfer\_fname

Description:

Customer file name Perl search pattern. It is processed as case insensitive.

Default:

**reg\_ex\_xfer\_fname = ${CID}\\_(\d{6})(.tif|.txt)(.7z.pgp|.pgp|.gpg)**

Name:

### reg\_ex\_img\_fname

Description:

The image/index data file name Perl search pattern. It is processed as case insensitive.

Default:

**reg\_ex\_img\_fname = ${CID}\\_(\d{6})(.tif|.txt)**

Name:

### reg\_ex\_encrypt\_ext

Description:

The recognized extension name for encrypted files.

Default:

**reg\_ex\_encrypt\_ext = (.pgp|.gpg)**

Name:

### reg\_ex\_zip\_ext

Description:

The recognized extension name for zipped files.

Default:

**reg\_ex\_zip\_ext = (.zip|.7z)**

Name:

### reg\_ex\_img\_ext

Description:

The recognized extension name for image files.

Default:

**reg\_ex\_img\_ext = (.tif|.tiff|.img)**

Name:

### reg\_ex\_idx\_ext

Description:

The recognized extension name for index files. Only applicable for image data file formats that use index files (currently ISR, and JHA).

Default:

**reg\_ex\_idx\_ext = (.txt|.xml)**

Name:

### reg\_ex\_file\_prefix

Description:

Search and replace Perl regular expression used to generate the Image/Index file pair association prefix entry in the file map table when the prefix for the Image/Index files is different.

For example, the image and index files for sjcb are named as sjcb\_tif\_*yymmdd*.tif and sjcb\_txt\_*yymmdd.*txt. To create the correct prefix to process the two files as a pair we could use the following regular expression:

/(sjcb\_)(txt\_|tif\_)(\w+)/$1$3/

The script will use the first and third string match to create the correct prefix entry in the map table to process the two files together. This option should be seldom used since most of the Image/Index files we receive have matching prefixes, and it is ignored for data file types that only use a single file.

Default:

**reg\_ex\_file\_prefix = undef**

Name:

### xfer\_file\_cnt

Description:

Defines the minimum number of expected customer files in the ftpbu directory.

Default:

**xfer\_file\_cnt = 1**

Name:

### img\_file\_cnt

Description:

Defines the minimum number of expected image files to process (after decryption and unzip if applicable).

Default:

**img\_file\_cnt = 1**

Name:

### request\_download

Description:

Boolean flag: 0 or 1. Enables customer data file ftp download.

Default:

**request\_download = 0**

Name:

### request\_xfer\_chk

Description:

Boolean flag: 0 or 1. Enables customer data file transfer check from file scanner.

Default:

**request\_xfer\_chk = 0**

Name:

### request\_decrypt

Description:

Boolean flag: 0 or 1. Enables customer file decryption.

Default:

**request\_decrypt = 1**

Name:

### request\_unzip

Description:

Boolean flag: 0 or 1. Enables customer file unzip.

Default:

**request\_unzip = 0**

Name:

### request\_chksum

Description:

Boolean flag: 0 or 1. Enables customer file checksum.

Default:

**request\_chksum = 0**

Name:

### request\_parse\_log

Description:

Boolean flag: 0 or 1. Creates and emails the parsing report.

Default:

**request\_parse\_log = 1**

Name:

### request\_img\_subdir

Description:

Boolean flag: 0 or 1. If true, creates a subdirectory within the directory pointed to by the *dir\_image* configuration option. The subdirectory will be named with the check posting date, in “*mmddyyyy*” format.

Default:

**request\_img\_subdir = 1**

Name:

### pwd\_unzip

Description:

Password to use if the zipped file is password protected.

Default:

**pwd\_unzip = undef**

Name:

### ftp\_host

Description:

FTP host to download customer files from.

Default:

**ftp\_host = undef**

Name:

### ftp\_usr

Description:

FTP login user name.

Default:

**ftp\_usr = undef**

Name:

### ftp\_pwd

Description:

FTP login password.

Default:

**ftp\_pwd = undef**

Name:

### ftp\_path

Description:

FTP directory containing customer files to download.

Default:

**ftp\_path = undef**

Name:

### xfer\_time\_interval

Description:

Customer file transfer check time period in minutes. Only used if the *request\_xfer\_chk* configuration option is set to true.

Default:

**xfer\_time\_interval = 120**

Name:

### maillist\_tag\_int

Description:

Mailing list configuration file tag containing the email addresses used for internal mail.

Default:

**maillist\_tag\_int = maillist1**

Name:

### maillist\_tag\_ext

Description:

Mailing list configuration file tag containing the email addresses used for external mail.

Default:

**maillist\_tag\_ext = maillist2**

Name:

### img\_fname\_format

Description:

Format used to create the output check images file names. The format string is comprised by a list of colon separated tokens that the script will substitute with the data extracted from the image data file. The colon character will be replaced with the “\_” underscore character. The following is a list and description of valid tokens recognized by the script:

acct\_num - Account number.

rout\_num - Routing number.

chk\_num - Check number.

chk\_amt - Check amount.

chk\_date - Check posting date.

img\_num - Image number. Currently supported only for JHA format datafiles.

item\_num - Item number.

trans\_code - Transit code.

aux\_on\_us - Auxiliary-On-Us code.

The following tokens are also accepted and the value following the equal sign will be used as follows:

prefix= - String to prefix the file name with. Default is the CID value.

postfixF= - String to appended to file name for front check image. Default Is “\_F”.

postfixB= - String to appended to file name for rear check image. Default Is “\_R”.

Default file name format:

**img\_fname\_format = prefix=${CID}:acct\_num:chk\_date:chk\_amt:chk\_num:postfixF=\_F:postfixB=\_B**

Name:

### err\_flag\_file

Description:

Name of the file used as an error flag. If a fatal error occurs during processing this file will be created.

Default:

**err\_flag\_file = $ENV{home\_env}/$ENV{keep\_dir}/${CID}\_chk\_img\_err.flag**

Name:

### maillist\_file

Description:

The name of the mailing list configuration file that contains the email address tags, referenced by the *maillist\_int* and *maillist\_ext* options.

Default:

**maillist\_file = $ENV{home\_env}/$ENV{keep\_dir}/maillist/${CID}\_mail.lis**

Name:

### chksum\_log\_file

Description:

The name of the checksum log file if file checksum is requested.

Default:

**chksum\_log\_file = $ENV{home\_env}/$ENV{keep\_dir}/cksum\_log/${CID}\_cksum.log**

Name:

### parsing\_log\_file

Description:

The name of the image parsing summary log file.

Default:

**parsing\_log\_file = $ENV{d\_dir}/prep/${CID}/log/tiff\_parsing.log**

Name:

### lock\_file

Description:

Process locking semaphore file name.

Default:

**lock\_file = $ENV{home\_env}/$ENV{keep\_dir}/${CID}\_chk\_img.lck**

# Control File Example:

**/home/control/fnbs\_image.control**

data\_file\_format = ISR #-- Image data file format.

dir\_ftpbu = /ftpbu/fnbsbu/DDA #-- File pickup directory relative to $ENV{d\_dir}

reg\_ex\_xfer\_fname = fnbs\_image\w\_\d{6}.zip.(pgp|gpg) #-- Transferred file name search pattern.

reg\_ex\_img\_fname = fnbs\_image\w\\_\d{6}.(tif|txt) #-- Image/Index file name search pattern.

request\_unzip = 1 #-- Boolean flag: 0 or 1

maillist\_file = fnbsms1\_mail.lis #-- Mailing list config file. Relative to

$ENV{home\_env}/$ENV{keep\_dir}/maillist.

maillist\_tag\_int = maillistInternal #-- Mailing list config file tag used for

internal mail.

maillist\_tag\_ext = maillist #-- Mailing list config file tag used for

external mail.

**/home/control/bkfn\_image.control**

data\_file\_format = X937 #-- Image data file format.

dir\_ftpbu = /ftpbu/bkfnbu/DDA #-- File pickup directory. Relative to $ENV{d\_dir}

dir\_backup = /download/bkfn #-- File backup directory. Relative to $ENV{d\_dir}

reg\_ex\_xfer\_fname = DataExport\d{6}.\d{2}.pgp #-- Transferred file name search pattern.

reg\_ex\_img\_fname = DataExport\d{6}.\d{2} #-- Image/Index file name search pattern.

img\_fname\_format = prefix=bkfn:acct\_num:chk\_amt:chk\_num:chk\_date #-- Output image file name format.

maillist\_tag\_int = maillist2 #-- Mailing list config file tag used for internal

mail.

maillist\_tag\_ext = maillist1 #-- Mailing list config file tag used for external

mail.

# Process Flow:

1. Set lock file.
2. Load configuration options from control file. If default control file exists or one is defined.
3. Validate and display loaded configuration options.
4. Retrieve internal and external mailing lists.
5. Prepares working directories for processing.
6. Download customer files from FTP site. If configured. Default = No
7. Check if the customer file scan has been completed. If configured. Default = No
8. Retrieve customer files.
9. Copy files to processing and archival directories.
10. Decrypt files. If configured. Default = Yes
11. Unzip files. If configured. Default = No
12. Perform file checksum. If configured. Default = No
13. Validate image files.
14. Process image files.
15. Send process summary notification. If configured. Default = Yes
16. Remove work files.
17. Remove lock file.

# Process Logs:

As default, the script generates the following log files under the directory ***/d/prep/{CID}/log/:***

***error\_report\_{YYYYMMDD\_HHMMSS\_DD}.txt*** (summary list of errors and files created)

***error\_log\_{YYYYMMDD\_HHMMSS\_DD}.txt*** (detail list of error messages)

***image\_parsed\_{YYYYMMDD\_HHMMSS\_DD}.txt*** (list of file names created)

***tiff\_parsing.log*** (process summary notifications archive)

The default log file directory (*/d/prep/{CID}/log/)*can be changed by updating the ***dir\_log*** configuration option in the control file.

The default name of the process summary notifications archive (*tiff\_parsing.log*) can be changed by updating the ***parsing\_log\_file*** configuration option in the control file.

Note: It is recommended to redirect the script’s STDOUT and STDERR to a disk file, perhaps under the same directory listed above, as in the examples shown in bold below:

If running from command line:

get\_chk\_image.pl -c svbt -f svbt\_image.control ***2>&1 | tee –a /d/prep/svbt/log/get\_chk\_image.log***

If running from cron:

get\_chk\_image.pl -c svbt -f svbt\_image.control ***2>&1 >> /d/prep/svbt/log/get\_chk\_image.log***

# Library Files:

The following is the list of the Perl modules used by the processing script:

get\_chk\_image.pl - Main processing script.

GetChkImg.pm - Base class module.

GetChkImgCOFF.pm - Processing module for COFF format image files.

GetChkImgISR.pm - Processing module for ISRetrieve format image files.

GetChkImgJHA.pm - Processing module for Jack Henry Associates format image files.

GetChkImgX937.pm - Processing module for X9.37 format image files.