FactorSoft™

- Release v4.7

# AutoCash

### Limitations on Maintenance Services

The FactorSoft™ application is intended for use in accordance with the standards and processes described within this documentation. Efforts to investigate and/or repair FactorSoft™ application or data integrity issues caused by activities or integrations outside of the intended use of the FactorSoft™ platform will be subject to the then-current Jack Henry Professional Services billable hourly rate.

Standard Maintenance Services (Technical Support) does not include the following:

> Investigation and Remediation of errors and data integrity issues caused, contributed to, or by any of the following:
>
> - a software program that was not originally provided by Jack Henry
>   - third-party automation, BOT/Screen Scraping technology, custom importers, or any other integration with FactorSoft™ that was not created by or in conjunction with Jack Henry.
> - any modification not provided by Jack Henry to the software or standard database schema
>   - the addition of custom database elements including triggers, stored procedures, tables, and columns
>   - the alteration of standard FactorSoft™ triggers, tables, columns, stored procedures and indexes
>   - the execution of T-SQL scripts resulting in changes to the data stored within the FactorSoft™ database
> - equipment, software, networks or any other infrastructure in the customer's environment that does not meet the minimum requirements described within the then-current FactorSoft™ product documentation

Please note that if you are exploring possibilities with third-party software providers or considering altering the FactorSoft database in any way, it is strongly recommended that you discuss your plans with the FactorSoft™ support team before making any commitments or changes. As your software partner, we may be able to help solve your business problem in a way that does not introduce risk, data corruption, or system instability.

# Table of Contents

# Working with AutoCash

The AutoCash module automates data entry of scanned images and data associated with posting payments The AutoCash module automates the three separate acts of Payment processing in FactorSoft: uploading and entering payment data, uploading images, and associating an image or images to the data for later recall.

AutoCash is configured for each client individually, with thresholds set for chargeback of payment overages and shortages. Overages, recoursed items, paid items, and unknown items update hold accounts specified for AutoCash in the Tables module.

Using AutoCash, third-party OCR systems can build cash batches with images to be imported in FactorSoft. AutoCash creates a cash batch, populates it with images and information, and then "wraps" the batch and delivers it to FactorSoft ready for processing.

> **SEPARATELY LICENSED ADD-ON MODULE**
>
> This feature is only available as a separately licensed upgrade.
> For more information, contact your Jack Henry$^{TM}$ representative at lendinginfo@jackhenry.com

## How it Works

This payment import method employs the fsChkImporter executable of FactorSoft for the direct creation of payment batches from third party applications and data sources.

This method is designed to interface with extremely high volume front end imaging, data entry and OCR/ICR systems. Each implementation of this solution is custom and the file produced depends on the file format produced by your lockbox vendor. Please provide your vendor with the three formats listed in Lockbox Index File Specification, so they can review which they may be able to produce. Once they produce a test, FactorSoft's technical team can help determine if its formatted correctly.

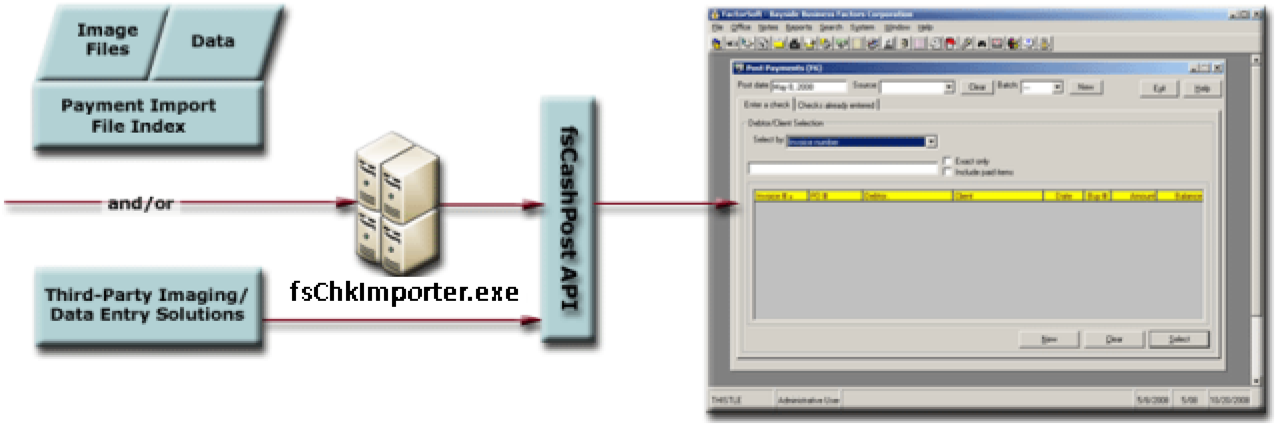
> **NOTE**
>
> AutoCash can be employed to import data into FactorSoft using the following methods.
>
> - Import a file provided from the feeding applications using Check Importer module – described in Processing a Data File via Check Importer module.
> - Interface directly from feeding application.

## Payment Import Criteria

AutoCash supports two methods for importing data and images to the Cash Posting module:

- **Parallel Data + Images** – allows for the importing of payment data, as well as corresponding images of the checks and backup (this method requires the Imaging module)
- **Serial Data + Images** – allows for the direct entry of payment images and data from third party high speed data entry, OCR/ICR providers.

Payment batches of data and images can be acquired from many lockbox vendors, or through the use of many popular 3rd party imaging/OCR applications. The Check Importer (fsChkImporter) is a standalone FactorSoft module that translates between the data/image capture method and the Engine. The current primary file layout is outlined in Lockbox Index File Specification



# Setting Up AutoCash

This section shows you how to set up FactorSoft to allow payment imports via AutoCash.

## System Preferences

The following System Preferences must be set in the **Data entry behavior**, **Cash posting**, **Auto cash folder**:

| Preference | Description |
|---|---|
| Matching rule | Select the rule to use when matching imported payment records to client/debtor outstanding invoice purchase records for payment:<br>• **ABA/Routing and three invoices must match**<br>• **Additionally, all payments must be equal or less than the invoice balance**<br>• **Additionally, all payments must be exactly equal to the invoice balance** |

The following System Preferences must be set in the **Data entry behavior**, **Cash posting**, **Field edits/actions** folder:

| Preference | Description |
|---|---|
| Automatically chargeback closed | Set this preference to **True** to automatically chargeback a closed invoice and display the adjustment amount in the **Write-off/chargeback** fields of the Post Payment window at time of cash posting. |
| Automatically chargeback non-factored | Set this preference to **True** to automatically chargeback a non-factored invoice and display the adjustment amount in the **Write-off/chargeback** fields of the Post Payment window at time of cash posting. |
| Automatically chargeback overage | Set this preference to **True** to automatically chargeback a user-defined payment overage and display the adjustment amount in the **Write-off/chargeback** fields of the Post Payment window at time of cash posting. |
| Automatically chargeback recoursed | Set this preference to **True** to automatically chargeback a recoursed invoice and display the adjustment amount in the **Write-off/chargeback** fields of the Post Payment window at time of cash posting. |
| Automatically | Set this preference to **True** to automatically chargeback a user-defined pay- |

| Preference | Description |
|---|---|
| chargeback shortage | ment shortage and display the adjustment amount in the **Write-off/chargeback** fields of the Post Payment window at time of cash posting. |

The following System Preferences must be set in the **Client rules/defaults**, **Defaults**, **Auto cash rules** folder. Note that the rules will only apply to newly added clients; they will not apply to existing clients.

| Preference | Description |
|---|---|
| C/B shortage % max | Enter the default shortage percentage rate at or over which an invoice is considered paid (i.e. closed). |
| C/B shortage $ max | Enter the default shortage dollar amount at or above which an invoice is considered paid (i.e. closed). |
| Overage hold account | Select the default hold account to be used for the automatic chargeback of overage amounts. |
| Overage maximum amount | Enter the overage amount at or below which an invoice is considered paid (i.e. closed). |
| Recourse action hold account | Select the hold account to be used for the automatic chargeback of recourse amounts. |
| Closed action hold account | Select the hold account to be used for the automatic chargeback of paid invoices. |
| Unknown action hold account | Select the hold account to be used for the automatic chargeback of unknown or unidentified amounts. |

The following System Preferences must be set in the **Identification/system constants**, **Performance**, **Scanning**:

| Preference | Description |
|---|---|
| Store scanned images in files | Set to **True** to store imported images in files. Note that fsCheckImporter will not store images in the database, so this preference must be set to True to import image files in AutoCash. |

The following System Preferences must be set in the **Data entry behavior**, **Imaging**:

| Preference | Description |
|---|---|
| Daily folder root path | Enter the root path for the daily folder used to store images. |

| Preference | Description |
|---|---|
| Use daily folders for images | Set to True to store images in the root path defined in the system preference **Data entry behavior, Imaging, Daily folder root path**. |

After defining the system preferences for imaging, the first daily must be created and the path name must be entered on the Image Folder Maintenance screen in the Administration module.

> NOTE
>
> The Image Folder Maintenance screen can be accessed by selecting System → Database Maintenance → Image Folder Maintenance.

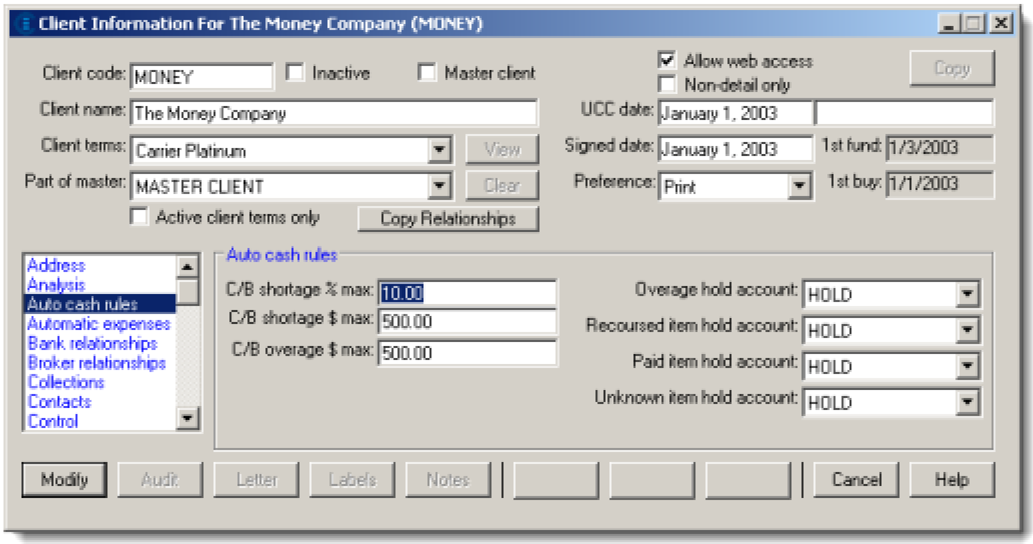
When entering the menu path, the server name should be used. For example, \\Bhm-QAAlpha\cadence-share\ images\2017\07\11.

The next daily folder is automatically created by the Engine during the Date Roll process.

## Configuring AutoCash Rules

Each client for whom you will import cash postings via AutoCash must have chargeback rules configured and G/L accounts specified into which the funds will be placed. AutoCash rules are defined on the Auto cash rules panel of the Client Information screen.

> NOTE
>
> System-wide defaults for these values can be set in System Preferences Client rules/defaults, Defaults, Auto cash rules folder.



Client Information For The Money Company (MONEY)

Client code: MONEY □ Inactive □ Master client

☑ Allow web access
□ Non-detail only

Client name: The Money Company

UCC date: January 1, 2003

Client terms: Carrier Platinum

Signed date: January 1, 2003 1st fund: 1/3/2003

Part of master: MASTER CLIENT

Preference: Print 1st buy: 1/1/2003

□ Active client terms only

Auto cash rules

C/B shortage % max: 10.00

C/B shortage $ max: 500.00

C/B overage $ max: 500.00

Overage hold account: HOLD

Recoursed item hold account: HOLD

Paid item hold account: HOLD

Unknown item hold account: HOLD

| Field | Description |
|---|---|
| C/B shortage % max | Enter the shortage percentage rate at or over which an invoice is considered paid (i.e. closed).<br>For example, entering a rate of 90% indicates that the client's invoice remains open if the payment received is less than 90% of the total invoice amount, and that it is closed if the payment received exceeds 90% of the total invoice amount. |
| C/B shortage $ max | Enter the shortage dollar amount at or above which an invoice is considered paid (i.e. closed).<br>For example, entering an amount of $200 indicates that invoices remain open if shortage amounts equal or exceed $200, and that they are closed if shortage amounts are lesser than $200. |
| C/B overage $ max | Enter the overage amount at or below which an invoice is considered paid (i.e. closed).<br>For example, entering an amount of $50 indicates that invoices are closed if overage amounts are equal to or less than $50, and that invoices payments exceeding $50 are posted to Hold Accounts. |
| Overage hold account | Select the hold account to be used for the automatic chargeback of overage amounts. |
| Recoursed item hold account | Select the hold account to be used for the automatic chargeback of recourse amounts. |
| Paid item hold account | Select the hold account to be used for the automatic chargeback of paid invoices. |
| Unknown item hold account | Select the hold account to be used for the automatic chargeback of unknown or unidentified amounts. |

## *Processing a Data File*

AutoCash imports data into FactorSoft automatically using the Check Importer module – described below:

### Processing a Data File via Check Importer Module

Follow these steps to import payments via the Check Importer.

1. Locate and run the fsChkImporter.exe file located in the FactorSoft Common folder (typically C:\FactorSoft).

   > **TIP**
   >
   > If an older version of FactorSoft is being used, the path may be displayed as C:\FactorSoft or C:\CADENCE.

   The Check Image Importer opens.

2. Click on the Database tab and enter the required data, as described below.

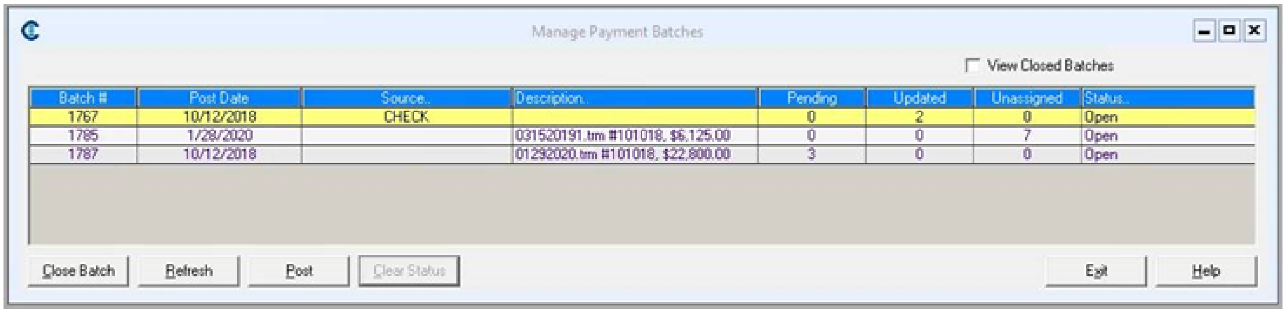
| Field | Description |
|---|---|
| Input | Enter the established directory structure for new, incoming import files. |
| Backup | Enter the established directory structure for outgoing, processed import files. |
| Errors | Enter the location for files that had errors and did not complete processing. Errors must be corrected, reprocessed, and moved manually. |
| Type | Select the database type into which the file is imported. |
| DB Name | Enter the database name into which the file is imported. |
| Server | Enter the name of the database server on which the database resides. |
| User | Enter the SQL login id for authentication into the server/database. |
| Password | Enter the SQL user password for authentication into the server/database. |
| Interval | Set the number of seconds that indicates how often the importer checks for new files to process. |
| Run unattended | Set this option as required. |
| Do not perform checksum test | Select this option to prevent check digit validation on the import file. |
| Format | Select the file format for the import. This format was developed for your organization during the implementation of AutoCash. |

3. Once setup is complete, the Engine will search the specified folder periodically, as set in the interval field, and extract data. When a fully downloaded file is detected, it is processed and moved to the specified outgoing directory. Batches can be sent one at a time throughout the day or together at end of day. When imported, batches will be labeled in FactorSoft as [remitplus]-[scannedbyuser]-[batchnumber]-[date].

> ### SEPARATELY LICENSED ADD-ON MODULE
>
> This feature is only available as a separately licensed upgrade.
> For more information, contact your Jack Henry$^{TM}$ representative at lendinginfo@jackhenry.com

4. Open Manage Payment Batches from the Office menu.

5. Select one of the imported payment batches and click **Post**.



| Batch # | Post Date | Source.. | Description.. | Pending | Updated | Unassigned | Status.. |
|---|---|---|---|---|---|---|---|
| 1767 | 10/12/2018 | CHECK | | 0 | 2 | 0 | Open |
| 1785 | 1/28/2020 | | 031520191.tim #101018, $6,125.00 | 0 | 0 | 7 | Open |
| 1787 | 10/12/2018 | | 01292020.tim #101018, $22,800.00 | 3 | 0 | 0 | Open |

6. Follow your normal posting procedures to post the imported data.

# How AutoCash Matches

1. AutoCash uses the ABA+Bank account number in the incoming file to gather a list of open invoice numbers and balances *from any debtor* who has paid from that back account before.

2. AutoCash then uses the AutoCash matching rules from the system preferences (**Data entry behavior**, **Cash posting**, **Auto Post**, **Matching rule**) to determine if a match is present:

   - **ABA/Routing and three invoices must match** - this is the loosest of the options. Of all the available invoice numbers from the select of open invoices (from #1 above), if any three numbers match, without any regard whatsoever to the invoice balances and payments, the specified payment amounts will be applied.

   - **Additionally all payments must be equal to or less than the invoice balance** - this option is a bit tighter than the first. In addition to the invoice numbers matching the subset selected in #1, the amounts being paid must be equal to or less than the open balance on the invoice.

   - **Additionally all payments must be exactly equal to the invoice balance** - this is the most conservative of the options. In addition to the invoice numbers matching the subset selected in #1, the amounts being paid must be *exactly* equal to the open balance on the invoice.

3. Based on the rules, if matches are found and a client debtor relationship can be found, then only the invoices for that client/debtor account are fetched and AutoCash attempts to post the payments (as the operator watches). As for the adjustments that *may* be required (the matching rules withstanding), the behavior is dictated by the Client Information Auto Cash Rules panel.

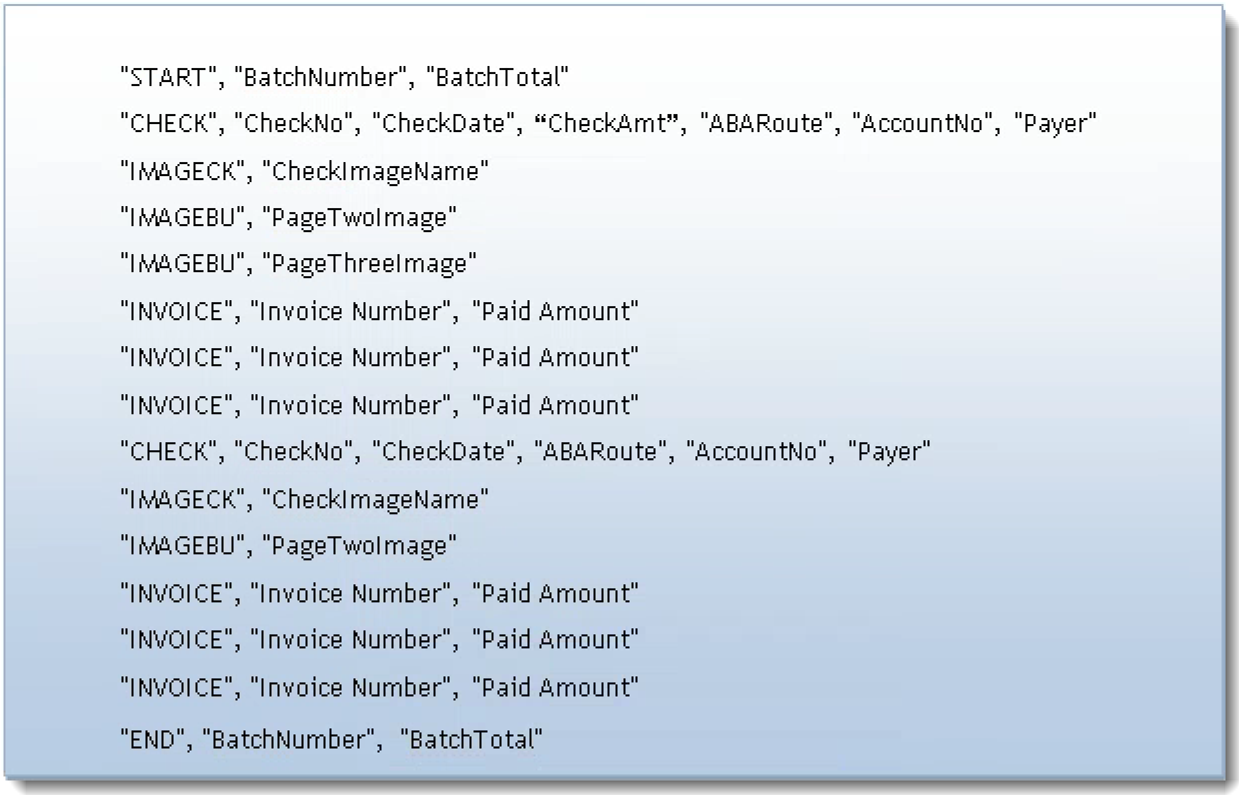
## Lockbox Index File Specification

There is an index file to accompany each payload or deliver of images and check data from a Lockbox vendor, to be imported into FactorSoft for cash posting. The index file follows a precise format in order to work properly:

- File is comma delimited, quote separated ASCII text
- Multiple check batches can be submitted in the same file by repeating the entire sequence from the START Batch to the END Batch designators
- The BatchNumber and the BatchTotal are passed as a reference only to the FSCashPost.DLL in FactorSoft
- Invoice Entries can be variable in number and are not required
- A transmission package—or payload—from the lockbox vendor consists of ZIP file containing a data file with a ".TRM" extension (e.g. LenderPay1.TRM) and all of the images referenced in the TRM Index file

> **NOTE**
>
> The ZIP file must be unzipped to process the file.

### *AutoCash Index File Structure*



```
"START", "BatchNumber", "BatchTotal"
"CHECK", "CheckNo", "CheckDate", “CheckAmt”, "ABARoute", "AccountNo", "Payer"
"IMAGECK", "CheckImageName"
"IMAGEBU", "PageTwoImage"
"IMAGEBU", "PageThreeImage"
"INVOICE", "Invoice Number", "Paid Amount"
"INVOICE", "Invoice Number", "Paid Amount"
"INVOICE", "Invoice Number", "Paid Amount"
"CHECK", "CheckNo", "CheckDate", "ABARoute", "AccountNo", "Payer"
"IMAGECK", "CheckImageName"
"IMAGEBU", "PageTwoImage"
"INVOICE", "Invoice Number", "Paid Amount"
"INVOICE", "Invoice Number", "Paid Amount"
"INVOICE", "Invoice Number", "Paid Amount"
"END", "BatchNumber", "BatchTotal"
```

Note that the CheckAmt includes the decimal point. For example, “600.05” and not “60005”.

### Example

For the example below, assume a $6,000 batch consisting of three checks and a $1000 batch consisting of a single check.

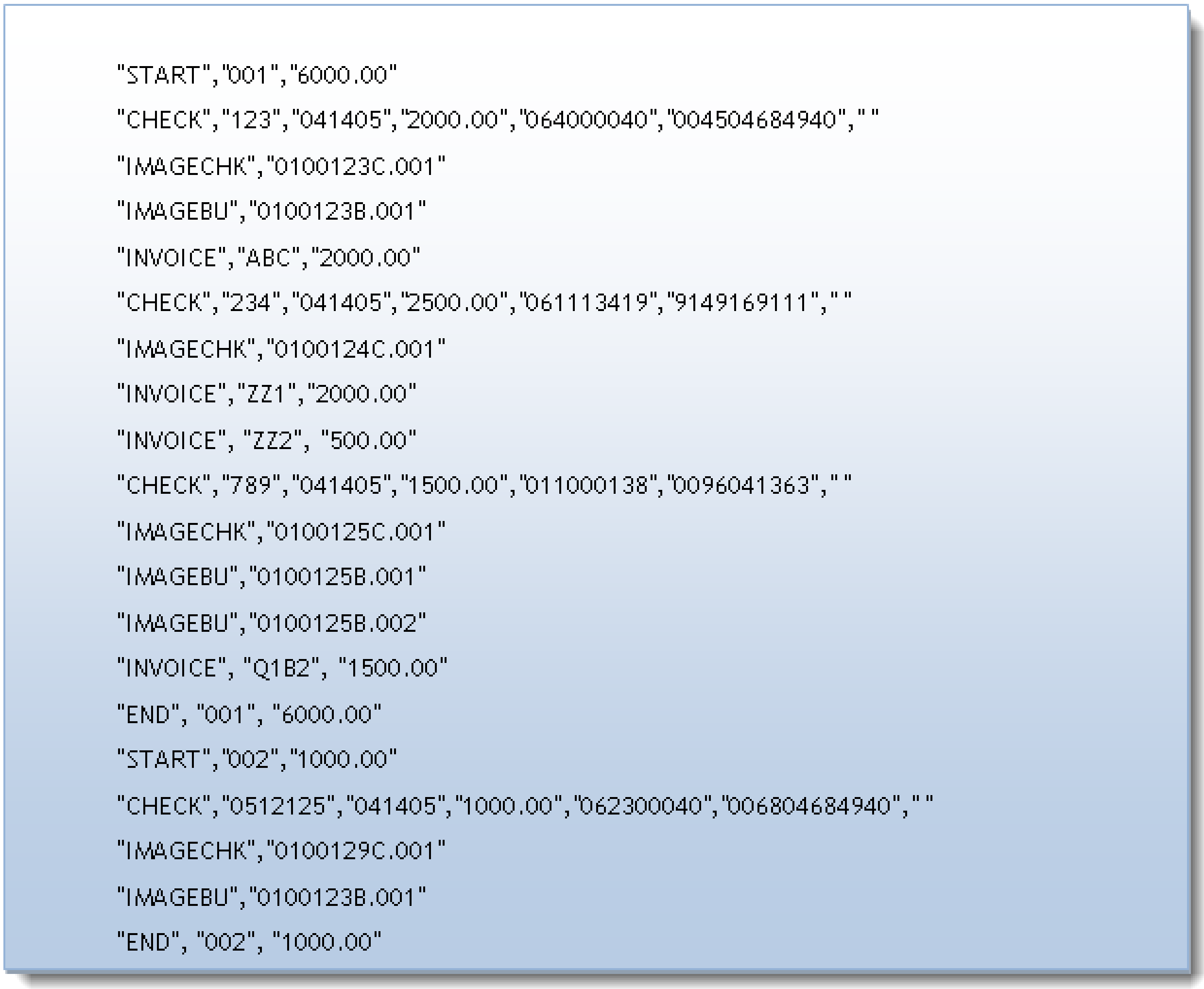
#### Batch 1

- Check 123 for $2000 pays invoice ABC for $2000.
- Check 234 for $2500 pays invoices ZZ1 for $2000 and ZZ2 for $500.
- Check 789 for $1500 pays invoice Q1B2 for $1500.

#### Batch 2

- Check 0512125 for $1000 with no remittance advice.

Given the above example, the file would look like:



```
"START","001","6000.00"
"CHECK","123","041405","2000.00","064000040","004504684940"," "
"IMAGECHK","0100123C.001"
"IMAGEBU","0100123B.001"
"INVOICE","ABC","2000.00"
"CHECK","234","041405","2500.00","061113419","9149169111"," "
"IMAGECHK","0100124C.001"
"INVOICE","ZZ1","2000.00"
"INVOICE", "ZZ2", "500.00"
"CHECK","789","041405","1500.00","011000138","0096041363"," "
"IMAGECHK","0100125C.001"
"IMAGEBU","0100125B.001"
"IMAGEBU","0100125B.002"
"INVOICE", "Q1B2", "1500.00"
"END", "001", "6000.00"
"START","002","1000.00"
"CHECK","0512125","041405","1000.00","062300040","006804684940"," "
"IMAGECHK","0100129C.001"
"IMAGEBU","0100123B.001"
"END", "002", "1000.00"
```

In the above file, the list of files that must be transmitted from the Lockbox vendor is:

The Index file from above with any name, but with an extension of ".TRM"

- 0100123C.001
- 0100123B.001
- 0100124C.001
- 0100125C.001
- 0100125B.001
- 0100125B.002
- 0100129C.001
- 0100123B.001

## AutoCash Bank of America File Format

### *File processing Notes for CHI-nnnnnn.csv*

General Notes:

1. Multiple batches may be contained within a file
2. Row 1 is the Header/Label Row of the header section of the file
3. Row 2 contains header information about the file.
    a. Row 2 Column A – should be passed as the Check batch source
    b. Row 2 Column G – should be passed as the Check Batch Date
    c. Row 2 Columns B, C, D, E, F, H , I, and J – should be ignored
4. Row 4 is the Header/Label Row for the detail section of the file
5. Row 5 to n contains the detail and image index data, but only lines coded as "CHECK" in column L should be read. Other indicators such as "Correspondence" should be ignored.
6. Possible images that can be passed are: Check, Invoice(s), Other Pages(s), Envelope(s). These should be passed in this order with Envelope image(s) always last.
7. Where pages have images of the back, the backs should be passed immediately following the page front
8. Each Batch in a file (indicated in column J) should be passed as a unique batch. For each batch, the amounts in column O should be totaled and passed to the API (Application Program Interface) as the batch total. As a validation, the detail lines should be totaled as they are passed, and must total the batch total calculated when beginning, else an integrity error is logged and the file contents are re-zipped and moved to the designated ERROR folder

### *Bank of America File Format*

| Col | Description | Sample Data | Req | Description |
|---|---|---|---|---|
| A | Detail Line Number | 1 | Ignore | |
| B | Envelope Number | 1 | Ignore | |
| C | Trans Number | 1 | Ignore | |
| D | Envelope ID | G-2266188 | Ignore | |
| E | Trans ID | G-2266188 | Ignore | |
| F | Trans Unique ID | TD648119d20083080000 | Ignore | |

| Col | Description | Sample Data | Req | Description |
|---|---|---|---|---|
| G | Lockbox ID | CHI-014906 | Ignore | |
| H | Date | 11/3/2014 | Pass as Check Date | |
| I | Time | 9:30 | Ignore | |
| J | Batch | 1 | Pass as Batch Number | Group Checks and break on Batch Number |
| K | Batch Item | 1 | Ignore | |
| L | Trans Source | Check | Read only lines coded as 'Check' | Ignore lines not coded as 'Check' |
| M | Group | | Ignore | Not currently used by BofA |
| N | Group Name | | Ignore | Not currently used by BofA |
| O | Amount | $329.25 | Pass as Check Amount | |
| P | ABA/RT | 74000010 | Pass as ABA number | |
| Q | Account Number | 714637444 | Pass as Account number | |
| R | Check number | 42731 | Pass as Check Number | |
| S | Number of Images | 15 | Use to validate the total number of images passed for this check | |
| T | Check Image Name | 8 | Pass \images\8.tif as the check image | |
| U | Check Back Image Name | 8b | Pass \images\8b.tif as the check image back if present | Optional, may not be present |

| Col | Description | Sample Data | Req | Description |
|---|---|---|---|---|
| V | Envelope Image Name | 1 | Pass \images\1.tif as the envelope image if present | If envelope images are present, they should be passed after all other images (Check front, Check back, Invoice Images with backs, other pages with backs, then envelopes and backs |
| W | Envelope Back Image Name | 1b | Pass \images\1b.tif as the envelope back image if present | If envelope images are present, they should be passed after all other images (Check front, Check back, Invoice Images with backs, other pages with backs, then envelopes and backs |
| X | Invoice Image Name | 32;33;34 | Pass \images\32.tif as the invoice image if present, then pass other backs and invoice images where present | Multiple pages are separated by semicolon |
| Y | Invoice Back Image Name | 32b;34b | Pass \images\32b.tif as the invoice image back if present. Backs should be integrated with fronts when backs are present | Multiple pages are separated by semicolon |
| Z | Other Pages Image Name (s) | 118;119;120;121 | Pass \images\118.tif as another page image if present, then pass other | Multiple pages are separated by semicolon |

| Col | Description | Sample Data | Req | Description |
|---|---|---|---|---|
| | | | backs and page images where present | |
| AA | Other Pages Image Back Names(s) | 119b;120b | Pass \images\119b.tif as the page back. Pass \images\120b.tif immediately following 120.tif (the front pages), etc. | Multiple pages are separated by semicolon |

## Treasury Cash Management (Regions Check) Import

The Regions check import process can be executed by the Engine. This import was originally executed using the fsChkImporter standalone executable. The Regions (Treasury Cash Management) format is available under the Engine version of the Check Importer. The Engine imports Zip files containing data file (.trm) and images (.tif) storing the data in the Precalc.XMLData table.

### *System Preferences*

To utilize this feature, the Folder Monitor Imports task must be enabled in the Engine, and the following System Preferences in the Identification/system constant, Interface parameters, Treasury Cash Mgmt import parameters folder must be completed:

| Preference | Description |
|---|---|
| Folder to Watch | Enter the fully qualified folder location in which to receive Regions check import files to be processed by the Engine. |
| Frequency (Minutes) | Enter the frequency in number of minutes with which the Engine will query the Folder to Watch to determine if there are Regions check import files to be processed.<br>**Note:** Minutes entered here are in addition to the settings set in the Default Priority column of the Engine Module. See the Administrator Guide for details on Default Priority. |
| Turn on logging | Set to **True** to write entries into the log file. |
| Continue Searching | Set to **False** to disallow the import process to continue if a check image file is |

| Preference | Description |
|---|---|
| | missing in the defined path. |

## *File Specifications*

The transmission file is a Zip file containing a data file (.TRM) extension and image files (tif). The .TRM file is comma delimited, quote separated ASCII Text and can contain multiple check batches within one file. The .TRM file is presented in the following format:

| Col 1 | Col 2 | Col 3 | Col 4 | Col 5 | Col 6 | Col 7 | Col 8 | Col 9 |
|---|---|---|---|---|---|---|---|---|
| "Start" | Batch Number | Batch Total | | | | | | |
| "Check" | CheckNo | CheckDate | CheckAmt | ABANo | AccountNo | Payer/ Debtor Name | ClientNo | ClientRefNo |
| “ImageCK” | CheckImage | | | | | | | |
| “ImageBU” | PageTwo Image | | | | | | | |
| "Invoice" | InvoiceNo | Paid Amount | Descr | ClientNo | ClientRefNo | | | |
| "End" | BatchNumber | BatchTotal | | | | | | |

## *Import Rules*

### Original Matching Logic:

**Rule #1**

If the file contains 3 invoices with the same Client/Debtor, the import does not require the user to post the entire check.

**Rule #2**

If the file does not contain at least 3 invoices with the same Client/Debtor, then still import the file and require the user to work and post the entire Check/Checks.

Displays informational message for user working the Check/Checks:“Can Not Determine Client”

New Matching Logic

**Rule #1**

If the import file contains ClientRefNo and/or ClientNo, then use new matching logic.

**Rule #2**

If the import file does not contain neither ClientRefNo nor ClientNo, then use original matching logic.

**Rule #3**

If Check's ClientRefNo and ClientNo do not match the Invoice/Invoice's ClientRefNo and ClientNo, then the Check will not automatically post and require user to work and post the entire Check.

Displays informational message for user working the Check: "Can Not Determine Client"

**Rule #4**

If the ClientRefno and/or ClientNo do not exists in the database, then still import the file and require the user to work and post the entire Check/Checks.

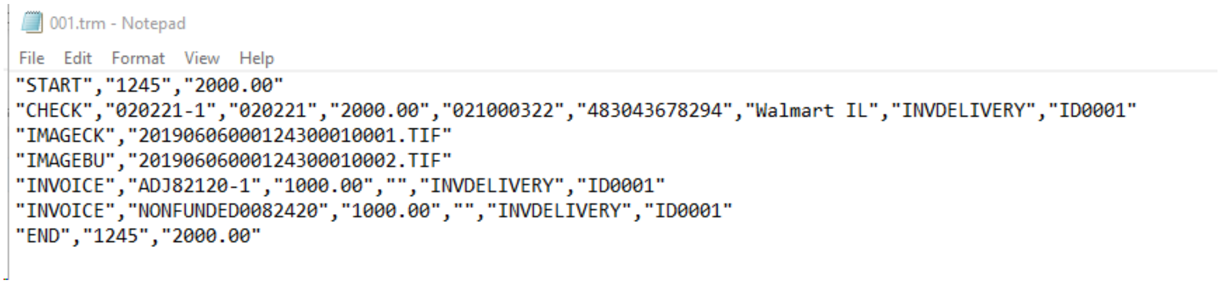
Displays informational message for user working the Check/Checks: "Can Not Determine Client"

**Rule #5**

If the Invoice/Invoices do not exist in the database, then the check will not automatically post and the user will be required to work and post the entire Check.

### *Sample Import File*

The below image is a sample of what each import file contains.



```
"START","1245","2000.00"
"CHECK","020221-1","020221","2000.00","021000322","483043678294","Walmart IL","INVDELIVERY","ID0001"
"IMAGECK","20190606000124300010001.TIF"
"IMAGEBU","20190606000124300010002.TIF"
"INVOICE","ADJ82120-1","1000.00","","INVDELIVERY","ID0001"
"INVOICE","NONFUNDED0082420","1000.00","","INVDELIVERY","ID0001"
"END","1245","2000.00"
```

## TRN Cash Management Import

The TRN Cash Management import process can be executed by the Engine. The Engine imports Zip files containing data file (.trn) or (.trm) and images (.tif) storing the data in the Precalc.XMLData table.

> **NOTE**
> For earlier version, prior to v4.1, the import will fail with Error Message “Process: File does not exist" if the zip (.zip) and data (.trn) file names do not match.

### *System Preferences*

To utilize this feature, the Folder Monitor Imports task must be enabled in the Engine, and the following System Preferences in the Identification/system constant, Interface parameters, TRN Cash Mgmt import parameters folder must be completed:

| Preference | Description |
|---|---|
| Folder to Watch | Enter the fully qualified folder location in which to import files to be processed by the Engine. |
| Turn on logging | Set to **True** to write entries into the log file. |
| Continue Searching | Set to **False** to disallow the import process to continue if a check image file is missing in the defined path. |
| Check Source Required | Set to **True**and if the ChkSource in the import file does not match value in the Check Source Table in the database or is missing, then the file will be rejected by CLMS.<br>Set to **False** to process original TRN Cash Management Format. |

### *File Specifications*

The transmission file is a Zip file containing a data file (.TRN) extension and image files (tif). The .TRN file is comma delimited, quote separated ASCII Text and can contain multiple check batches within one file. The .TRN file is presented in the following format:

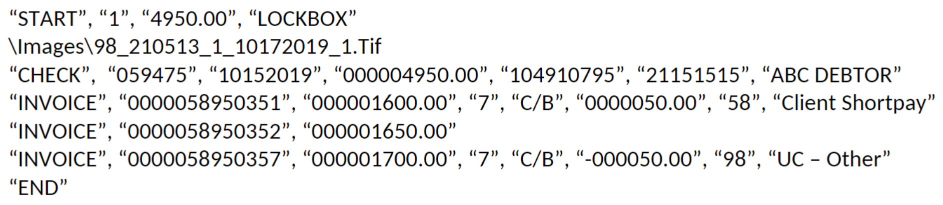
| Column 1 | Column 2 | Column 3 | Column 4 | Column 5 | Column 6 | Column 7 | Column 8 |
|---|---|---|---|---|---|---|---|
| "START" | Batch Number | Total Amount of Checks | | | | | |
| \Images\ | Image File Name | | | | | | |

| Column 1 | Column 2 | Column 3 | Column 4 | Column 5 | Column 6 | Column 7 | Column 8 |
|---|---|---|---|---|---|---|---|
| "CHECK" | Check Number | Date | Check Amount | Account Number | ABA Number | Debtor Name | |
| "INVOICE" | Invoice Number | Payment Amount | Write-off/Chargeback Code | Write-off/Chargeback Description | Short Payment Amount | Short Payment Amount | Dispute Code |
| "END" | | | | | | | |

- If import file contains Writeoff/Chargback Code and Description, then they must match the CBCodesType codes in the database or the file should fail to import.
- If the import file contains Dispute Key and Dispute Code, then they must match the Dispute/Ineligibility/No Buy Codes Table in the database, or the file should fail to import.

### *Sample Import File*

The below image is a sample of what each import file contains.



```
“START”, “1”, “4950.00”, “LOCKBOX”
\Images\98_210513_1_10172019_1.Tif
“CHECK”,  “059475”, “10152019”, “000004950.00”, “104910795”, “21151515”, “ABC DEBTOR”
“INVOICE”, “0000058950351”, “000001600.00”, “7”, “C/B”, “0000050.00”, “58”, “Client Shortpay”
“INVOICE”, “0000058950352”, “000001650.00”
“INVOICE”, “0000058950357”, “000001700.00”, “7”, “C/B”, “-000050.00”, “98”, “UC – Other”
“END”
```

## Wells Fargo Check Import

The Wells Fargo Check import process can be executed by the Engine. The Engine imports ZIP files storing the data in the Precalc.XMLData table. FactorSoft will process both Consolidated or Batch by Batch zip files with or without a .VAK file record. Image files included may be Black and White or Color.

**Batch by Batch** import file type process will contain Action: A (Add New Record). ZIP file will contain a transaction data file that are pipe-delimited (.dat) and image files (.tiff). Value-added Keying(VAK) – is included in the transaction data file and has four standard fields: Invoice amount, Invoice number, remitter name and check date.

**Consolidated** import file type process consists of a fixed length, space delimited ZIP file containing a transaction data file that are plain ASCII Text Files (Flat file), image files (.tiff) and a VAK File (fields.dat). Transaction file is 182 characters in length and VAK, if included, is 133 characters in length. ZIP file should process with or without the separate VAK File. Process will be reflected on the Import History report

Once imported, the data is accessed via the **Manage Payment Batches** screen to review, post, and process.

### *System Preferences*

To utilize this feature, the Folder Monitor Imports task must be enabled in the Engine, and the following System Preferences in the Identification/system constant, Interface parameters, Wells Fargo Check import parameters folder must be completed:

| Preference | Description |
|---|---|
| Auto Post Matching Parameters | Specifies if discounts should be applied to matching.<br>• Full Check Amount<br>• Discount Check Amount<br>• Both Full and Discount Check Amounts |
| Folder to Watch | Enter the fully qualified folder location in which to import files to be processed by the Engine. |
| Use Auto Post Matching | Set to **True** to use Auto Post Matching which automatically matches imported checks to existing invoices. |
| Import File Type | Set Import File Type - Batch by Batch, Consolidated |

### *File Specifications*

| Field Type | Column | Length | Description |
|---|---|---|---|
| **Batch by Batch** - The transmission file is a ZIP file containing a data file (.dat) extension and image files (tif). The .dat file is pipe delimited and can contain multiple check batches within one file. The .dat file is presented in the following format: | | | |
| Action Type | 1 | 1 | **A** Add (new record).<br>**C** Change. A change record is generated when a batch is updated at the lockbox site (for example, when a check is missed and then included with the updated batch). If this occurs, delete all records from the original batch. The batch with the C action code replaces the original batch.<br>**D** Delete a previously sent batch. If this occurs, delete all |

| Field Type | Column | Length | Description |
|---|---|---|---|
| | | | records in the specified batch.<br>Files do not include records with multiple types of action codes. For example, a file with an add record won't include any change or delete records. If a file includes a delete record, that is the only record in the file.<br>The type portion of the corresponding zip file name indicates the value for this field.<br>• If the type portion of the zip file name is SLBW or ALBW, this field is A for all records in the transaction data file.<br>• If the type portion of the zip file name is RLBW, this field is C or D for all records in the transaction data file. |
| Deposit Date | 2 | 8 | Deposit Date in YYYYMMDD format. |
| Lockbox Number | 3 | 7 | Lockbox number associated with the transaction. |
| Site ID | 4 | 3 | Code associated with the Lockbox Operations Center.<br>PH Phoenix, AZ<br>SF Fremont, CA<br>LA Los Angeles, CA<br>DV Denver, CO<br>MCO Orlando, FL<br>ATL Atlanta, GA<br>IL Chicago, IL<br>ORD Chicago, IL<br>DM Des Moines, IA<br>BWI Baltimore, MD<br>MA Boston, MA<br>SP Minneapolis/St. Paul, MN<br>NYC New York, NY<br>CLT Charlotte, NC<br>PD Portland, OR<br>PHL Philadelphia, PA<br>DL Dallas/Irving, TX<br>SL Salt Lake City, UT<br>SE Seattle, WA |
| Depository Account | 5 | 10 | Account number of your Wells Fargo account where the check was deposited. |

| Field Type | Column | Length | Description |
|---|---|---|---|
| Batch | 6 | 8 | System-generated number assigned to a group of transactions. |
| Transaction Number | 7 | 4 | System-generated number assigned to each transaction. This number links the CHK, INV, and WHT records for a transaction. |
| Record Type | 8 | 3 | **CHK** Check record - **INV** Invoice record - **WHT** White paper record. |
| Sequence Number | 9 | 4 | System-generated number assigned to each record type for a transaction. |
| Check Amount | 10 | 12 | For check records, the amount of the check. For invoice and white paper records, the field is null. For correspondence-only transactions, the field is 0.00. |
| Serial Number | 11 | 16 | For check records, the serial number on the check. The field is null for the following:<br>• Check records for correspondence-only transactions<br>• Invoice records<br>• White paper records |
| Check Account Number | 12 | 20 | For check records, the remitter's account number on the check. The field is null for the following:<br>• Check records for correspondence-only transactions<br>• Invoice records<br>• White paper records |
| Check Date | 13 | 8 | For check records, the date on the check in YYYYMMDD format. If no data is provided, the field is null. |
| Check RTN | 14 | 9 | For check records, the remitter’s routing/transit number on the check. The field is null for the following:<br>• Check records for correspondence-only transactions<br>• Invoice records<br>• White paper records<br><br>**Note** Any leading zeros in the RTN are removed. For example, an RTN of 091000019 would be included in the file as 91000019. |

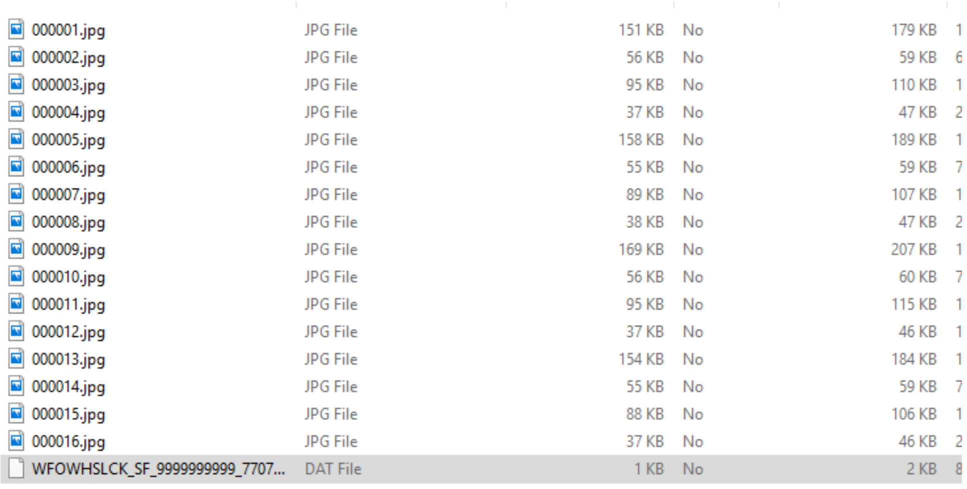
| Field Type | Column | Length | Description |
|---|---|---|---|
| Remitter Name | 15 | 100 | For check records, the remitter's name on the check. If no data is provided, the field is null. Field is free-form, and field length varies based on item data, typically not exceeding 100 characters. |
| Invoice Number | 16 | 100 | For invoice records, the invoice number captured at the lockbox site. For check and white paper records, the field is null. Field is free-form, and field length varies based on item data, typically not exceeding 100 characters. |
| Invoice Amount | 17 | 100 | For invoice records, the invoice amount captured at the lockbox site. For check and white paper records, the field is null. Field is free-form, and field length varies based on item data, typically not exceeding 100 characters. |
| Custom Data | 18 | 100 | Custom field captured at lockbox site. If no data is provided, the field is null. Field is free-form, and field length varies based on item data, typically not exceeding 100 characters. |
| Front Image File Name | 19 | 64 | Name of the front image file for the item. The field is null for the following:<br>• Check records for correspondence-only transactions<br>• Invoice records<br><br>**Note** Image file numbers begin at 000001 for the first image in each file and increment for additional images in the file. Since image file numbers reset to 000001 for each file, you need to use an archival strategy that allows for this and avoids overwriting image files from prior files with image files from new files. |
| Rear Image File Name | 20 | 64 | Name of the rear image file (if any) for the item. If there is no rear image, the field is null. |
| **Consolidated** - consists of a fixed length, space delimited ZIP file containing a transaction data file that are plain ASCII Text Files (Flat file), image files (.tiff) and a VAK File (fields.dat). | | | |
| Filler | 1 | 9 | Space-filled Filler |
| Site ID | 2 | 3 | Code associated with the Lockbox Operations Center. |

| Field Type | Column | Length | Description |
|---|---|---|---|
| | | | 602 Phoenix, AZ<br>510 Fremont, CA<br>626 Los Angeles, CA<br>720 Denver, CO<br>407 Orlando, FL<br>770 Atlanta, GA<br>312 Chicago, IL<br>773 Chicago, IL<br>319 Des Moines, IA<br>443 Baltimore, MD<br>617 Boston, MA<br>612 Minneapolis/St. Paul, MN<br>212 New York, NY<br>336 Charlotte, NC<br>541 Portland, OR<br>215 Philadelphia, PA<br>469 Dallas/Irving, TX<br>435 Salt Lake City, UT<br>425 Seattle, WA |
| Filler | 3 | 2 | Space-filled. |
| Lockbox Number | 4 | 10 | Lockbox number associated with the transaction. |
| Deposit Date | 5 | 8 | Deposit date in YYYYMMDD format. |
| Zeros | 6 | 4 | Always 0000 (four zeros). |
| Batch Number | 7 | 8 | Btach Number |
| Filler | 8 | 3 | Space-filled. |
| Transaction ID | 9 | 3 | Transaction sequence number within a batch. |
| Filler | 10 | 2 | Space-filled. |
| Check ID | 11 | 3 | Check sequence number within a transaction. |
| Filler | 12 | 2 | Space-filled. |
| Fields Sequence ID | 13 | 3 | Document sequence number within a transaction.<br>**Note:** If this number exceeds 999, the thousands digit is populated in position 56. |

| Field Type | Column | Length | Description |
|---|---|---|---|
| Document Type | 14 | 2 | **CK** Check record - **IN** Invoice record , envelope, or any other white paper record. |
| Check Amount | 15 | 14 | Amount of the check. Only populated if the document type is **CK**. Includes a decimal point and two decimal places. |
| Check Serial Number | 16 | 12 | Serial number of the check. Only populated if the document type is **CK**. |
| Filler | 17 | 1 | Space-filled. |
| Routing/Transit Number | 18 | 20 | Routing/transit number (RTN) on the check. Only populated if the document type is **CK**.<br>**Note:** Any leading zeros in the RTN are included. For example, an RTN of 091000019 would be included in the file as 091000019. |
| Account Number | 19 | 20 | Account number on the check. Only populated if the document type is **CK**. |
| Image Path Name | 20 | 54 | Name and location of the image file. |

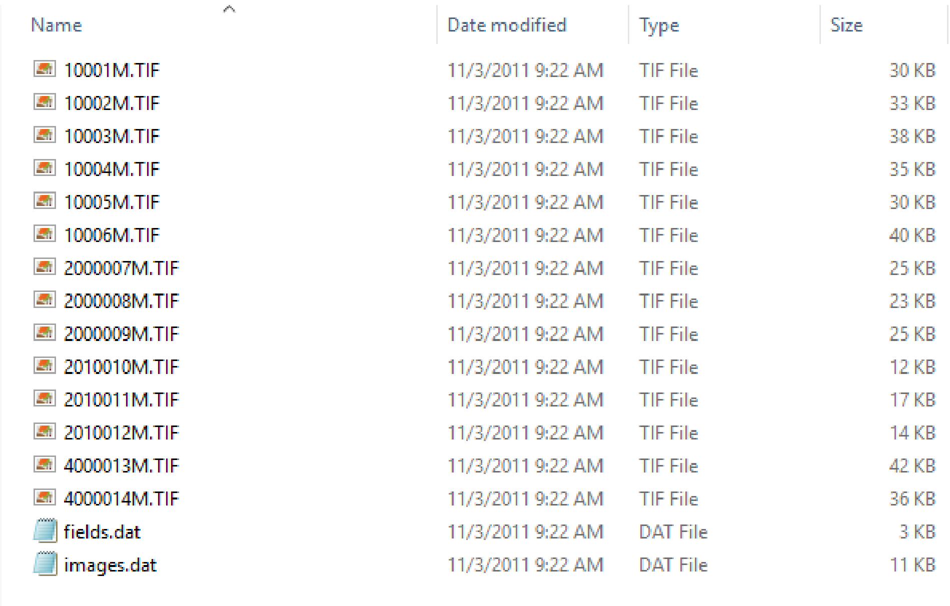
### *Sample Import File*

The below image is a sample of what each .zip and .dat import file contains.



| | | | | | |
|---|---|---|---|---|---|
| 000001.jpg | JPG File | 151 KB | No | 179 KB | 1 |
| 000002.jpg | JPG File | 56 KB | No | 59 KB | 6 |
| 000003.jpg | JPG File | 95 KB | No | 110 KB | 1 |
| 000004.jpg | JPG File | 37 KB | No | 47 KB | 2 |
| 000005.jpg | JPG File | 158 KB | No | 189 KB | 1 |
| 000006.jpg | JPG File | 55 KB | No | 59 KB | 7 |
| 000007.jpg | JPG File | 89 KB | No | 107 KB | 1 |
| 000008.jpg | JPG File | 38 KB | No | 47 KB | 2 |
| 000009.jpg | JPG File | 169 KB | No | 207 KB | 1 |
| 000010.jpg | JPG File | 56 KB | No | 60 KB | 7 |
| 000011.jpg | JPG File | 95 KB | No | 115 KB | 1 |
| 000012.jpg | JPG File | 37 KB | No | 46 KB | 1 |
| 000013.jpg | JPG File | 154 KB | No | 184 KB | 1 |
| 000014.jpg | JPG File | 55 KB | No | 59 KB | 7 |
| 000015.jpg | JPG File | 88 KB | No | 106 KB | 1 |
| 000016.jpg | JPG File | 37 KB | No | 46 KB | 2 |
| WFOWHSLCK_SF_9999999999_7707... | DAT File | 1 KB | No | 2 KB | 8 |

Consolidated zip file shown below:



| Name | Date modified | Type | Size |
|---|---|---|---|
| 10001M.TIF | 11/3/2011 9:22 AM | TIF File | 30 KB |
| 10002M.TIF | 11/3/2011 9:22 AM | TIF File | 33 KB |
| 10003M.TIF | 11/3/2011 9:22 AM | TIF File | 38 KB |
| 10004M.TIF | 11/3/2011 9:22 AM | TIF File | 35 KB |
| 10005M.TIF | 11/3/2011 9:22 AM | TIF File | 30 KB |
| 10006M.TIF | 11/3/2011 9:22 AM | TIF File | 40 KB |
| 2000007M.TIF | 11/3/2011 9:22 AM | TIF File | 25 KB |
| 2000008M.TIF | 11/3/2011 9:22 AM | TIF File | 23 KB |
| 2000009M.TIF | 11/3/2011 9:22 AM | TIF File | 25 KB |
| 2010010M.TIF | 11/3/2011 9:22 AM | TIF File | 12 KB |
| 2010011M.TIF | 11/3/2011 9:22 AM | TIF File | 17 KB |
| 2010012M.TIF | 11/3/2011 9:22 AM | TIF File | 14 KB |
| 4000013M.TIF | 11/3/2011 9:22 AM | TIF File | 42 KB |
| 4000014M.TIF | 11/3/2011 9:22 AM | TIF File | 36 KB |
| fields.dat | 11/3/2011 9:22 AM | DAT File | 3 KB |
| images.dat | 11/3/2011 9:22 AM | DAT File | 11 KB |