micros

Restaurant Enterprise Series

Settle-to-File Credit Card Driver for 3700 POS Version 4.x

August 15, 2008

MD0003-076

Installation and Setup

This section contains installation and setup instructions for the Version 4.x release of the Settle-to-File (STF) Credit Card Driver . The release version is available on the MICROS web site Product Support page.

In This Section...

Installation

Settle-to-File (STF) is a custom Credit Card Driver that performs credit card settlement to a local disk file. It does NOT perform an on-line settlement to a credit card processor.

During settlement, the driver formats the credit card records contained in a batch and writes them to the designated local file. Correctly formatted batches will be approved and any errors will be reported.

Site Requirements

Before installing the STF Credit Card Driver on the res3000 system, the following configuration items should be considered:

- The installed version of 3700 POS should be Version 3.2 or higher.

Files Included

The STF driver is for settlement only. The following files are installed with this driver:

\Micros\RES\POS\Bin\CaSTF.dll

\Micros\RES\POS\etc\CaSTF.cfg

\Micros\RES\POS\Bin\CaSTF.hlp

\Micros\RES\POS\Bin\CaSTF.cnt

Software Installation Instructions

The installation of credit card drivers is now separate from the res3000 software. After each installation of res3000 software — whether it is a general release, service pack, or hotfix — you MUST re-install the site's requisite credit card drivers.

1. Make sure all current batches have been settled. MICROS recommends installing a new driver before the site opens for the day. Doing so ensures that all CA/EDC transactions have been settled to their current version.
2. Download the **STF419614.zip** file from the MICROS web site. Copy this file to your RES Server's temp folder and unzip the files.
3. Shutdown all MICROS applications from the MICROS Control Panel.
4. Copy the following files to the correct folder locations listed below:
   - **CaSTF.dll** to **\Micros\Res\Pos\bin**
   - **CaSTF.cfg** to **\Micros\Res\Pos\etc**
   - **CaSTF.hlp** to **\Micros\Res\Pos\bin**
   - **CaSTF.cnt** to **\Micros\Res\Pos\bin**
5. Turn on the res3000 system from the MICROS Control Panel.

Driver Setup Instructions

1. Create a new CA/EDC record in *POS Configurator* | *Devices* | *CA/EDC Drivers.*
2. Enter the **Driver Code** — *STF.*
3. No other setup is required for this driver.

CA/EDC should be operational. A few test transactions should be done to ensure all is working correctly.

File Format

Filename

Bxxxxxxxxx.txt where **x** = nine-digit batch number (e.g., B000000012.txt)

Location

\micros\res\pos\ccbatch on the RES Server

> ***Note*** *The default destination directory will be named **CCBatch** under the POS directory. The destination directory will be changeable via the **Option** registry key, using the value **CCBatchPathname**. An example of the format for the pathname is: **..****ccbatch***

Format

- The file will be an ordinary ASCII file containing a Header Record, an Open Record, one or more Payment Records, and a Close Record.
- Each record will consist of a series of comma-separated fields.
- The fields may vary in length, depending on the data contained therein.
- Each record will end with a CR-LF sequence (newline or \n).
- The contents of each field will be in quotes, e.g., “field contents”.
- Empty fields will also be quoted, like this: ,””,
- Any quote character that occurs within a field’s data will be replaced with two consecutive quotes in the file record, e.g., “here comes a quotee: ““notice that it is doubled”
- Numeric data will be expressed as strings, e.g., an opcode value of 10 will be expressed as the string, “10”.
- All credit card information is contained in this file unmasked.
- The goal will be to make the file importable into Excel.

Header Record Contents

The Header Record will contain all system and merchant configuration items that originate from the Configurator.

Field No.	Description	Format
1	Record Type = “H”	Single character. Must be “H”.
2-11	Ten consecutive user-configurable system numeric fields	Integer strings, up to 9 numerals (e.g., “123456789”).
12-21	Ten consecutive user-configurable system string fields	Character strings, up to 25 positions (e.g., “Example System String”).
22-31	Ten cpmsecutive user-configurable merchant numeric fields	Integer strings, up to 9 numerals (e.g., “123456789”).
32-41	Ten consecutive user-configurable merchant string fields	Character strings, up to 25 positions (e.g., “Example Merchant String”).

Open Record Contents

The Open Record will contain all of the fields that the Credit Card Server (CCS) can send to a settlement driver in its own Open Record. These fields will be as follows, in the order given:.

Field No.	Description	Format
1	Record Type = “O”	Single character. Must be “O”.
2	Batch Number	Integer string ranging from “1” through “9999999” (e.g., “24”).
3	Gross Batch Amount[1]	Cash amount ranging from “0” to “999999999999” (e.g., “5843”).
4	Batch Sales Amount[1]	Cash amount ranging from “0” to “999999999999” (e.g., “5979”).
5	Batch Credits Amount[1]	Negative cash amount ranging from “0” to “–999999999999” (e.g., “–136”).
6	Business Date	YYMMDD (e.g., “030825”).
7	Batch RVCS[2]	Integer string ranging from “1” through “9999999” (e.g, “1”).
8	Number of Batch Records	Integer ranging from “1” through “9999999” (e.g, “3”).

Field No.	Description	Format
9	Number of Batch Sales	Integer ranging from “1” through “9999999” (e.g, “2”).
10	Number of Batch Credits	Integer ranging from “1” through “9999999” (e.g, “1”).
11	Create Date/Time	YYMMDDHHMMSS (e.g., “030825144033”).
12	Earliest Date/Time from Payment Records	YYMMDDHHMMSS (e.g., “030825143753”).
13	Latest Date/Time from Payment Records	YYMMDDHHMMSS (e.g., “030825144019”).

Payment Record Contents

The Payment Record will contain fields that the CCS can send to a settlement driver in its own Payment Record. These fields will be as follows, in the order given:.

Field No.	Description	Format
1	Record Type = “P”	Single character. Must be “P”.
2	Payment Record Number	Integer (e.g., “1”).
3	Account Entry Mode	Single character, “0” for manual, “1” for swiped.
4	Card Number	Character string, 20 positions, left-justified, space-filled on right (“e.g., “6019440005202522 “).
5	Card Expiration Date	MMYY (e.g., “0920”).
6	Track 2 Data	Character string, up to 40 positions (e.g., “;6019440005202522=20900000000001000?”).
7	Track 1 Data	Character string, up to 80 positions (e.g., “%B6019444000520252522^^20090000000110000000?”).
8	Tender Number – Configured object number of the tender associated with this payment.	Integer ranging from “1” to “9999999” (e.g., “300”).
9	Terminal – Configured object number of workstation where payment occurred.	Integer ranging from “1” to “9999999” (e.g., “99”).

Field No.	Description	Format
10	RVC Number[2] – Configured object number of the RVC where this payment occurred.	Integer ranging from “1” to “9999999” (e.g., “1”).
11	Guest Check Number	Integer ranging from “1” to “9999” (e.g., “1956”).
12	Check Employee – Configured object number of the employee performing the payment.	Integer ranging from “1” to “9999999” (e.g., “101”).
13	Date/Time of CC Final[3]	YYMMDDHHMMSS (e.g., “030825143752”).
14	Base Payment[1]	Cash amount ranging from “-999999999999” to “999999999999” (e.g., “5028”).
15	Actual Charged Tip Payment[1]	Cash amount ranging from “-999999999999” to “999999999999” (e.g., “00”).
16	Cashback Payment[1]	Cash amount ranging from “-999999999999” to “999999999999” (e.g., “00”).
17	Total Payment[1]	Cash amount ranging from “-999999999999” to “999999999999” (e.g., “5028”).
18	Immediate Payment Flag	Single character, either “1” for True or “0” for False.
19	Number of Authorization Records to follow. See below.	Single character, “1”, “2”, or “3”.
20	CC Start Date	MMYY (e.g., “0809” or ““ (null)).
21	CC Issue Number	Two characters or ““ (null).
22	Debit Flag	Sincle character, either “1” for True or “0” for False.

File Management

The STF Driver does not manage the batch files it creates. It is the responsibility of the installer to manage the files created by this driver.

ReadMe First Version 4.x.xx.xxxx

This section contains a comprehensive guide to the new features, enhancements, and revisions included in the Version 4.x release of the Settle-to-File (STF) Credit Card Driver.

In This Section...

What's New

A new feature is defined as one that provides capabilities that were not available in previous versions of the application.

New Features Summarized

The following table summarizes the new features included in this version:

Feature	Page
Settle to Pipe	4

New Features Detailed

Settle to Pipe

The STF (Settle to File) credit card driver provides a method of passing settlement data to a customer application using comma delimited records written to a disk file. Due to the security issues created by writing sensitive data to a disk file, even for a very short time, the Settle-to-Pipe feature is designed to send the settlement records through a direct channel between the driver and a third-party application.

Using the Feature

This section some use cases for the Settle-to-Pipe feature:

Example 1

A third-party application uses an encryption package to encrypt the settlement records before writing them to disk. The encrypted settlement files are then transmitted during the normal overnight polling.

Example 2

A third-party application establishes an encrypted communication channel with a corporate host. The settlement records are passed across this channel to the corporate host for settlement.

How It Works

In Settle to Pipe mode, when the STF driver receives a batch open request from the credit card service, it creates a pipe which can be used to write the settlement batch to the pipe mode client. The STF driver launches the pipe mode client application, passing the batch sequence number to the third-party application as a command line parameter.

Notes

The STF driver runs under the Local System account and does not have access to the computer desktop. The client application should either be a console mode application or an application which is able to run without being attached to the desktop.

Windows also limits the ability of applications executed by the Local System account to be authenticated on a remote computer, which may limit access to resources across a network.

After the driver verifies that the application has been launched it will wait for the application to open the pipe before transmitting settlement records. A configurable timeout controls the amount of time the driver will wait for the client application. Once the client application has opened the pipe, the driver will transmit one record at a time.

After each record has been transmitted, the driver will wait for the record to be processed before writing the next record. Once the driver has written the batch close record it will close the pipe. Once the driver closes the pipe, it will signal the credit card service that the batch has been successfully settled. A timeout at any point before the batch close record is received by the client will be treated as a failed settlement attempt and the batch will be left unsettled.

When multiple batches are being settled, the driver will wait for up to 60 seconds for the client application to terminate before creating a new pipe and re-launching the application for the next batch.

The driver expects the client to return the total number of records, including batch overhead records, read as the application exit code. The driver will compare the exit code to the number of records sent and will report any mismatch in the batch settlement report. A mismatch in the record count will not cause a batch settlement to fail. If the driver sends the batch close record to the client it will mark the batch as settled.

No changes were made to the STF record format. The driver will write each record to the pipe followed by a line feed character. The existing record order, batch header record ("H"), batch open record ("O"), batch detail records ("P"), followed by a batch close record ("C") will be used.

Registry Configuration

The Settle to Pipe feature must be configured via the registry. The following registry values control the settle to pipe feature. All are stored under the driver's `Option` key:

- **PipeModeClient (REG_SZ)** – The path to the client executable to be launched by the driver. This can be an absolute path or a path relative to the *RES\POS\ETC* directory. Pipe mode is disabled if the value is not present. For example:

  D:\Test\PipeClient.exe (client is located in D:\Test)

- **SecurePipeClient (REG_DWORD)** – The driver can, optionally, test the application before it is launched to verify that the application is signed and that the certificate used to sign the application is trusted by the server PC. Value is 0 or 1 (default 0). The driver will use Microsoft Authenticode technology to verify the digital signature of the application.

  If enabled, the driver will verify that the application has been digitally signed and that the certificate used to sign the application was issued by a trusted certificate authority (CA). Root certificates for commercial CA's such as Verisign, Thawte, or Comodo are already installed on the server PC by Microsoft. It is also possible for the customer to install their own root certificate and use that certificate to sign the client application. More information about Authenticode and signing applications can be found at http://www.tech-pro.net/code-signing-for-developers.html.

- **ConnectionTimeout (REG_DWORD)** – The time, in seconds, that the driver will wait for the client application to open the pipe (default 30).

- **TransmitTimeout (REG_DWORD)** – The time, in seconds, that the driver will wait for the client application to read each settlement record after it is written to the pipe (default 10).

Follow these steps to configure the registry settings for Settle to Pipe:

1. Open Regedit to *\\HKLM\SOFTWARE\MICROS\Common\CCS\Drvrcfg\DrvrX* (where X= the STF driver sequence number).

2. Under the STF driver sequence number, make sure that you have a key called **[Option]**. If not, create one.

3. Under the **[Option]** key, create a STRING value called **PipeModeClient**. The name of this STRING value will differ depending upon the name of the executable created by the third-party.

4. (Optional) Under the **[Option]** key, create the following **DWORD** values:

   - SecurePipeClient
   - ConnectionTimeout
   - TransmitTimeout

   These values should be used if the site requires configuration other than the default configuration. For example, if the site requires more than the 30 second default **ConnectionTimeout** value, then the user would need to create a **ConnectionTimeout** option key. However, if 30 seconds is sufficient, then no optional key is needed.

   A description of each of these values, as well as the default configurations are listed above.

Command Line Options (Automatically Passed by the Driver)

The driver formats the command line for the client application with two parameters. These parameters are:

- **nnnnnnnnn** – This is the sequence number of the batch being transmitted.
- **pipe_name** – The pipe name has the general form \\.\pipe\nnnn where nnnn is a random number. Windows allows a pipe to be opened and processed as if it is a file so the client application can handle pipe_name internally as if it were a file name.

The following is a sample pipe command line:

```
D:\Test\PipeClient.exe 000000005 \\.\pipe\34395
```

What's Enhanced

An enhancement is defined as a change made to improve or extend the current functionality. To qualify as an enhancement, the change must satisfy the following criteria:

- The basic feature or functionality already exists in the previous release of the software.
- The change adds to or extends the current process. This differs from a revision (i.e., a bug fix) which corrects a problem not caught in previous versions.

Enhancements Summarized

The following table summarizes the enhancements included in this version:

Feature	Page

Enhancements Detailed

Title

Text.

What's Revised

A revision is defined as a correction made to any existing form, feature, or function currently resident in the 3700 POS software. To qualify as a revision, the change must satisfy the following criteria:

- The basic form, feature, or functionality must be a part of the previous version of the software.
- The change must replace or repair the current item or remove it from the application.

Revisions Summarized

The following table summarizes the revisions included in this version:

Feature	CR ID	Page
Driver Not Logging with RES 4.x	N/A	9
Not Inheriting Verbosity from Credit Card Server	N/A	9
Timeout Values are in Milliseconds Rather than Seconds	N/A	9

Revisions Detailed

Driver Not Logging with RES 4.x

CR ID #: N/A

Previously, the STF driver was not logging messages to the 3700d.log when run with RES version 4.x. This has been corrected.

Not Inheriting Verbosity from Credit Card Server

CR ID #: N/A

Previously, the STF driver was not inheriting its verbosity level from the Credit Card Server. Now, the driver will inherit the verbosity setting from the Credit Card Server, unless the verbosity of the STF driver is configured to be higher. In that situation, the driver will use the higher of the two settings.

Timeout Values are in Milliseconds Rather than Seconds

CR ID #: N/A

Previously, the timeout settings for the STF driver were incorrectly recorded in miliseconds, when they should be in seconds. This has been corrected.