Card Loader

> *Previously AA32-GPayments Card Loader.pdf*

GPayments' Card Loader and Signer/Verifier (Card Loader) is a Java-based application that can be used for signing, verifying and transmitting cardholder registration messages to GPayments' Registration API Server.

The Registration API Server requires XML messages to be signed prior to transmission. The Card Loader can be used to sign and verify these messages.

The application provides two mechanisms for sending XML messages to the registration server. The messages can either be manually scheduled by the administrator or copied to the **In Tray** directory for automatic transmission.

This section is designed for administration personnel who are responsible for uploading signed cardholder enrolment information to the issuer authentication system.

Installation

System Requirements

Operating System: Sun Solaris 10 (using X-Windows), Windows 10, Windows 8.1, Windows 7 (service pack 1)

Minimum Hardware: UltraSPARC II 400MHz 128M RAM, Intel PIII 500MHz 128M RAM

Java Runtime Environment: JRE 1.6.22 [1] or later

Disk Space: ~1.8Mb for the application itself. Allow enough space for the installation of JRE and creating signed XML files.

JDK/JRE

The application requires the installation of the latest release of JDK or JRE 1.7 or 1.8. JRE and JDK (both openJDK and Oracle JDK) can be freely downloaded from Sun Microsystems at http://java.sun.com/. JDK and JRE must be installed with the default settings. Follow the on screen installation instructions for JRE or JDK to complete the installation.

Installing the Application

- Unzip the contents of the package into a temporary directory
- To start the command line installation program: for UNIX, run setup.sh and for Windows, run setup.bat
- Follow the prompts and select a destination directory. If the specified directory contains a previous version of the application, it will be upgraded.

Running the Application

The application must be run in graphical mode in order to access all the available functionality. A subset of tasks can be run from command line.

Running the Application in Graphical Mode

- From the destination directory run the following command:

  For Solaris:

  - run **start.sh**

  For Windows:

  - run **start.bat**

> **Note**
>
> **Note:** The default **Password** is **123456**. You should change this password as soon as possible.

Running the Application from the Command Line

- Edit start.sh for UNIX or start.bat for Windows and add the required command line parameters at the end of the start file.

Command line options:

```
[-s\|-v] [-p <password>] [<input_file>] [<output_file>]
```

-s: Sign the input XML file specified by input_file

-v: Verify the signature of the input file.

-p: Provide the password in the command line. If not specified the user will be prompted to enter a password.

input_file:
The input XML file which needs to be signed or verified. The application prompts the user for a valid file if not specified.

output_file:
The outcome of signing is stored in an output file specified by this parameter. The application prompts the user for an output file if not specified.

> **Note**
>
> **Note:** The default password is **123456**. You should change this password as soon as possible.

Signing Large XML Files

The default Java heap size is 64MB. Even though a larger amount of physical and virtual memory might be available on your system, your Java process is not allowed to use the extra memory, which may result in an out of memory exception when signing or verifying larger XML files.

In order to increase the amount of memory available to your Java process add:

-Xms< min_size > -Xmx< max_size > switches when invoking the application.

Example 1:

java –Xms256m –Xmx256m [class path and jar file here]

Sets the amount of heap memory available to the application to 256MB.

Example 2:

java –Xms128m –Xmx512m [class path and jar file here]

Sets the minimum amount of memory available to the application to 128MB and allows the heap size to increase up to 512MB if necessary.

Card Loader Application Interface (GUI)

Logging in for the First Time

KeyStore Password dialog

- Use the default password **123456** to login to the application for the first time.

> **Note**
>
> For security reasons, you should change this as soon as possible and select a non-trivial password.

> **Note**
>
> For security reasons, you must **create a new set of keys** when you log into the application for the first time. This ensures that the keys are unique to your organization. You must also export the certificate and send it to your provider once you have created a new set of keys. This ensures that they have the correct public key in order to validate the messages signed by your organization.

The Main Window

The main window shows the list of currently scheduled and in progress jobs (In Tray interface) or the list of completed jobs (Sent Items).

The File, Card and Tools menus provide options for managing cards and certificates. These options are also available via toolbar buttons.

To view details for an item

- Double click on an item in the **In Tray** or **Sent Items**.

Item menu

To view a list of available functions for an item

- Right click on an item.

Scheduling a Job

To manually schedule a job:

- From the **File** menu, point to **New** and select **New Card...** or **New Notification…**, as appropriate.
- Select the file you wish to upload and set a date and time.

  If you do not specify a date and time, the scheduler will set it to current date and time and will run the job as soon as possible.

  The Card Loader sends one message at a time, which means if multiple jobs are scheduled to run at the same time only one is run and others will wait in the queue until the current job is complete.

  You can also copy registration API XML files directly to the **In Tray** directory (as specified in **Tools /Options** window). Files copied here will be sent to the registration server as soon as possible.

Schedule a New Card Request

Schedule a New Notification Request

Signing an XML Message

To sign an XML message:

- From the **Tools** menu, point to **Signer** and select **Card Request Signer** or **Notification Request Signer**, as appropriate.

Open dialog for selecting registration XML file

- Enter the **File name** of a valid registration XML file and click **Open**.

Save dialog for entering output file name

- Enter a **File name** for the output file and click **Save**.

> **Note**
>
> **Input** and **output** files must be different.

The application validates the syntax of the input file against the registration API DTD and will only sign the message if it is a valid XML API message.

Message signed dialog

Cancel Signing an XML Message

To cancel the signing process:

- Select **Stop** from the **Card** menu.

Verifying an XML Message

To verify an XML Message:

- From the **Tools** menu, point to **Verifier** and select **Card Request Verifier** or **Notification Request Verifier**, as appropriate.

Open dialog for selecting XML file to verify

You can view and change the following options:

- **In Tray directory (for input files):** Specify the input directory. The XML files are kept in this directory until corresponding job is complete. You can directly copy the XML messages into this directory in order to mark them for immediate upload.
- **Sent Items directory (for output files):** Specify the output directory. Once a job is complete, the corresponding XML file is moved to the **Sent Items** directory.
- **Log directory (for output files):** Specify the log directory. The outcome of a finished job (either completed or failed) is stored in a log file.

  For added security, CardLoader encrypts critical data such as card name and card number using a hardcoded key and decrypts this data for displaying.
- **KeyStore directory:** Specify the KeyStore directory. KeyStore directory should point to where the KeyStore and cacerts files are stored. The default is 'KeyStore'.

- **Scheduler time interval**: (minutes) Define how often the scheduler should check for new files in the In Tray directory.
- **Reschedule failed jobs in:** Specify a delay (in hours) for rescheduling those jobs that fail due to connectivity errors.

  To disable rescheduling of failed jobs, enter zero (0).
- **Number of cards per file for splitting:** Specify a limit for the number of cards sent in each registration message. The application will break input files, which contain more cards than the specified limit, into smaller chunks.

  To disable file splitting, enter zero (0).
- **Schedule interval between two uploads** (seconds)

  Where multiple files are uploaded to the system at the same time, this parameter will determine the time period between the scheduled start times of each of the file upload jobs.

  To disable this interval, enter zero (0).
- **Wait period between two uploads** (seconds)

  Where multiple files are uploaded to the system at the same time, this parameter will determine a minimum period between the scheduled start times of each of the file upload jobs and takes priority over the **Schedule interval between two uploads value**, where this is not **0**.

  To disable this parameter, enter zero (0).
- **Sign files copied to In Tray –** select this check box to automatically sign files copied to the In Tray
- **Compress files before upload –** select this check box to compress messages before sending them to the registration server. This will improve upload time and save some bandwidth.

Tools > Options > Connection tab

You can view and change the following connection options:

- **Registration API server:** Specify the full URL of the registration server here, e.g.: https://127.0.0.1:8080/registration
- **SSL Protocol:** Select the SSL protocol from the dropdown: **TLSv1, TLSv1.1 or TLSv1.2**.

> **Warning**
>
> For security reasons, only TLSv1.1 and higher should be used.

Proxy Settings

- **Enable proxy server:** select this check box and enter **Proxy host** and **Proxy port** if you do not directly connect to Internet and need to go through a proxy server.
- **Authentication:** select this check box and enter the **User name** and **Password** if your proxy server requires authentication.

> **Note**
>
> The application supports basic proxy authentication and basic LDAP authentication.

Tools > Options >Email Settings tab

You can view and change the following email settings:

- **Enable email notification service:** This option must be enabled before you can specify other email settings
- **Sender email address:** Email address of the sender of the notification message.
- **Administrator email address:** The email address of the recipient of the notification message. This should normally be the administrator in charge of the card upload process.

  You can use a group email address here in order for the messages to be sent as many people as required.
- **Mail server host:** Enter the URL of the outgoing email server (SMTP).
- **Mail server username and password:** Enter a username and password for connection to the email server. This must be a valid account on your mail server.
- **Notification flags:** select from the list of available notification flags. You may choose a notification to be sent when a job is scheduled (rescheduled), starts, is completed or when a job fails.

SSL

Card Loader comes packaged with a comprehensive list of public/commercial certificate authorities. This means that the application can successfully connect to the registration server as long as the server certificate is signed by one of these trusted CAs.

If you use a private CA such as the one established by your own organization, you need to import the CA certificate to the list of trusted certificates in order for Card Loader to be able to establish connection with the server.

To do this, go to the / KeyStore directory and use the "keytool" command as follows:

```
\$ keytool -import [-v] [-noprompt] [-trustcacerts] [-alias <alias>]
s[-file <cert_file>] [-keypass <keypass>] [-keystore <keystore>]
[-storepass <storepass>] [-storetype <storetype>] [-provider
<provider_class_name>] ...
```

e.g. Importing certificate file (GPayments CA.cer)

```
\$ keytool -import -trustcacerts -alias "gpaymentsca" -file "GPayments CA.cer"
-keystore cacerts
```

- Enter your login password as the KeyStore password.

---

1. JRE is not necessarily backward compatible. An application that works with JRE 1.7.0_72 does not necessarily work with JRE 1.7.0_10, for example. Make sure that you have installed the correct version of JRE. If you have installed multiple versions of JRE/JDK on the host system, be sure that the correct version is used to start the application.