Introduction

Version 9.0.4:2

ActiveAccess is a modular, multi-issuer authentication system for eCommerce transactions. It supports 3D Secure authentication for American Express SafeKey, Discover ProtectBuy, JCB J/Secure, Mastercard Identity Check, Verified by Visa (VbV) and Visa Secure in a single system, allowing issuing banks to participate in any, or all, of the supported authentication schemes. ActiveAccess supports two-factor authentication for 3D Secure based on devices such as SMS, email, OOB, Decoupled Authentication and one-time password (OTP) devices.

Product Architecture

This section describes the product architecture and its logical components. Understanding the logical units of the application should help you with designing the actual implementation of the product to meet the deployment and security requirements of your organisation.

In this guide we use the term **server** for any software component that can be accessed via a client application, in a standard client/server architecture. To avoid any confusion we use the term **physical server** when referring to the hardware itself.

Internal Components

Main Components

The main components of ActiveAccess are:

- Access Control Server
  - Verify Enrolment Server
  - Payer Authentication Server
  - Authentication Server
  - Challenge Server
  - RMI Server
  - AHS Client
  - Rules Engine
  - External Messaging Adapter
  - Risk Engine Adapter
  - Out of Band Authentication Adapter
  - Decoupled Authentication Adapter
- Administration Server
- Registration Server
- Whitelisting Server
- Database Server

Server components are implemented as servlets that can be deployed to any one of the commercial application servers supported by ActiveAccess.

Access Control Server (ACS)

ACS is the authentication component of the system. It provides a facility allowing communication and messaging with other authentication components during an authentication.

ActiveAccess ACS supports the **3-D Secure protocol**.
3-D Secure 1 is an authentication standard for online eCommerce transactions introduced by Visa and adopted by Mastercard, JCB, American Express and Diners Club International.
3-D Secure 2 is an update of the 3-D Secure 1 authentication standard, created by EMVCo to support app-based authentication and integration with digital wallets, as well as a frictionless authentication flow.

Verify Enrolment Server (3DS1)

Default port: Determined by the application server

Default path: Refer to the table in Access Control Server

Protocol: HTTP/HTTPS

Inbound connections: Directory server

Outbound connections: Database server

Other requirements: Must be able to access the HSM

The verify enrolment server is used in the 3-D Secure 1 processes. The verify enrolment server consumes VEReq messages and generates VERes messages accordingly.

Note that any changes to the fully qualified URL of the verify enrolment server must be reported to the 3-D Secure 1 providers in order to update the corresponding directory servers.

Payer Authentication Server (3DS1)

Default port: Determined by the application server

Default path: /acs/pa

Protocol: HTTP/HTTPS

Inbound connections: Cardholder's browser

The payer authentication server is used for cardholder authentication in the 3-D Secure process. The cardholder is redirected to the authentication server by the merchant plug-in during the 3-D Secure process. The authentication pages are stored in the database and served via the authentication server itself. The payer authentication server is responsible for the processing of the PAReq and generation of PARes message pair in the 3-D Secure 1 process.

Authentication Server (3DS2)

Default port: Determined by the application server

Default path: Refer to the table in Access Control Server

Protocol: HTTP/HTTPS

Inbound connections: Directory server

Outbound connections: Database server, Directory server

Other requirements: Must be able to access the HSM

The authentication server is used for cardholder authentication in 3-D Secure 2 process. The authentication server consumes AReq messages and generates ARes messages accordingly. For each AReq that is received, if challenge is required, the authentication server generates and RReq message to notify the directory server of the result of challenge, then consumes and RRes message in response to RReq.

Note that any changes to the fully qualified URL of the authentication server must be reported to the 3-D Secure 2 providers in order to update the corresponding directory servers.

Challenge Server (3DS2)

Default port: Determined by the application server

Default path: /acs/ca

Protocol: HTTP/HTTPS

Inbound and outbound connections: Cardholder's browser, 3DS SDK app

The Challenge server is used in the 3-D Secure 2 process. The challenge server consumes CReq messages and generates CRes messages accordingly.

RMI Server

Default port: 4241 and 4242

Protocol: JRMP (TCP) [1]

Inbound connections: Other ActiveAccess RMI servers, MIA

Outbound connections: Database server, Other ActiveAccess RMI servers

Other requirements: Must be able to access the HSM

The RMI server is used to synchronise a cluster of ActiveAccess servers. This is mainly to notify other ActiveAccess servers of changes in the settings of the cluster or to apply settings to multiple ActiveAccess servers from a single ActiveAccess administration interface.

RMI server is used when ActiveAccess components are deployed on multiple servers or multiple ActiveAccess servers are used for load balancing.

AHS Client (3DS1)

Default port: N/A

Default path: N/A

Protocol: HTTPS

Inbound connections: None

Outbound connections: Authentication history server, Database server

Other requirements: Must be able to access the HSM

In accordance with 3-D Secure 1 specification, a copy of transaction response (PARes) must be sent to the card scheme's designated server known as the Authentication History Server (AHS). The AHS client is responsible for sending the transaction record (PATransReq) to the designated AHS server.

Note that some 3-D Secure providers may not require or support an AHS.

Rules Engine

Default port: None

Default path: None

Protocol: None

Inbound connections: None

Outbound connections: Database server

Other requirements: None

The Rules engine is used for applying business rules for checking authentication requests processed or transparently authenticated by local or remote authentication servers.

Authentication exemption rules for local and remote authentication servers are:

- Soft Launch List
- Merchant Whitelist
- Merchant Watchlist
- Location Watchlist
- Domestic & International Transaction Amount Threshold
- Stand-In Transaction Threshold (remote authentication model)

Registration enforcement rules for local authentication servers are:

- Amount Threshold
- Merchant Blacklist

External Messaging Adapter

Default port: N/A

Default path: N/A

Protocol: HTTP/HTTPS

Inbound connections: N/A

Outbound connections: Centralised Authentication and Authorisation Service (CAAS), Database server

Other requirements: Must be able to access the HSM

The external messaging adapter manages the messaging requirements for connecting ActiveAccess to the issuers' remote systems.

Risk Engine Adapter

Default port: N/A

Default path: N/A

Protocol: N/A

Inbound connections: N/A

Outbound connections: RESTful RBA adapters

Other requirements: N/A

The Risks engine is used for applying risk rules for checking authentication requests processed or transparently authenticated by local or remote authentication servers. In an authentication, a challenge may be necessary because the transaction is deemed high-risk, e.g. above certain thresholds.

For risk assessment, ACS sends/receives proper data elements to/from risk assessment systems via middleware.

There are two types of risk adapters available:

- Native API version of Risk Adapter
- Restful API version of Risk Adapter

Out of Band (OOB) Authentication Adapter

Default port: N/A

Default path: N/A

Protocol: N/A

Inbound connections: N/A

Outbound connections: RESTful OOB adapters

Other requirements: N/A

The OOB is challenge activity that is completed outside of, but in parallel to, the 3-D Secure flow.

ActiveAccess performs Out Of Band (OOB) challenges through OOB adapters. OOB adapters connect the existing OOB authentication system with ActiveAccess. During 3-D Secure 2 challenge flows where OOB authentication is required, the ACS will trigger the external OOB process, perform interactions with the cardholder via the OOB adapters.

For this purpose, the ACS communicates with the existing OOB system via a middleware. This middleware is the OOB adapter. The OOB adapter can either be loaded locally by the ACS (Native API) or communicated with via HTTP calls (REST API).

Decoupled Authentication Adapter

Default port: N/A

Default path: N/A

Protocol: N/A

Inbound connections: N/A

Outbound connections: RESTful Decoupled adapters

Other requirements: N/A

Decoupled Authentication is authentication activity that is completed outside of, but in parallel to, the 3-D Secure flow. ActiveAccess performs Decoupled Authentication challenges through Decoupled Authenticator adapters. Decoupled Authenticator adapters connect the existing Decoupled authentication system with ActiveAccess. During 3-D Secure 2 flows where Decoupled authentication is required, the ACS will trigger the external Decoupled Authenticator process, and perform interactions with the cardholder via the Decoupled Authenticator adapters. For this purpose, the ACS communicates with the existing Decoupled Authentication system via a middleware. This middleware is the Decoupled Authenticator adapter. The Decoupled Authenticator adapter can either be loaded locally by the ACS (Native API) or communicated with via HTTP calls (REST API).

Administration Server

The management and reporting utility for the system is the administration server used by administrative users.

Default port: Determined by the application server

Default path: /mia/

Protocol: HTTP/HTTPS

Inbound connections: Administrator browser (Issuer's admin staff and internal admin staff)

Outbound connections: Database server, Registration Server, RMI Server

Other requirements: Must be able to access the HSM

The administration server is used by technical and issuer and helpdesk staff who are in charge of operations, maintenance and customer support. The administration server allows access to various system and business settings, and cardholder information, transactions, reports and logs.

Registration Server

A web service providing issuers the ability to enrol cardholders in real-time with the authentication schemes.

Default port: Determined by the application server

Default path: /registration/

Protocol: HTTP/HTTPS

Inbound connections: Issuer's registration software (such as Card Loader utility), Administration server

Outbound connections: Database server

Other requirements: Must be able to access the HSM

The registration API is used by issuers to register cardholders (pre-registration and final registration models).

Whitelisting Server

A web service providing issuers the ability to see/remove cardholders' whitelisted merchants in real-time via the Administration Server and add cardholder's whitelisted merchants in real-time via the Access Control Server.

Default port: Determined by the application server

Default path: /whitelisting/wl/api/merchant/

Protocol: HTTP/HTTPS

Inbound connections: Issuer’s software (such as issuer's website), Administration server

Outbound connections: Database server

Other requirements: Must be able to access the HSM

The Whitelist API is used by ActiveAccess and Issuers to manage cardholders' whitelisted merchants.

Database Server

Default port: 1521

Default path: N/A

Protocol: TCP

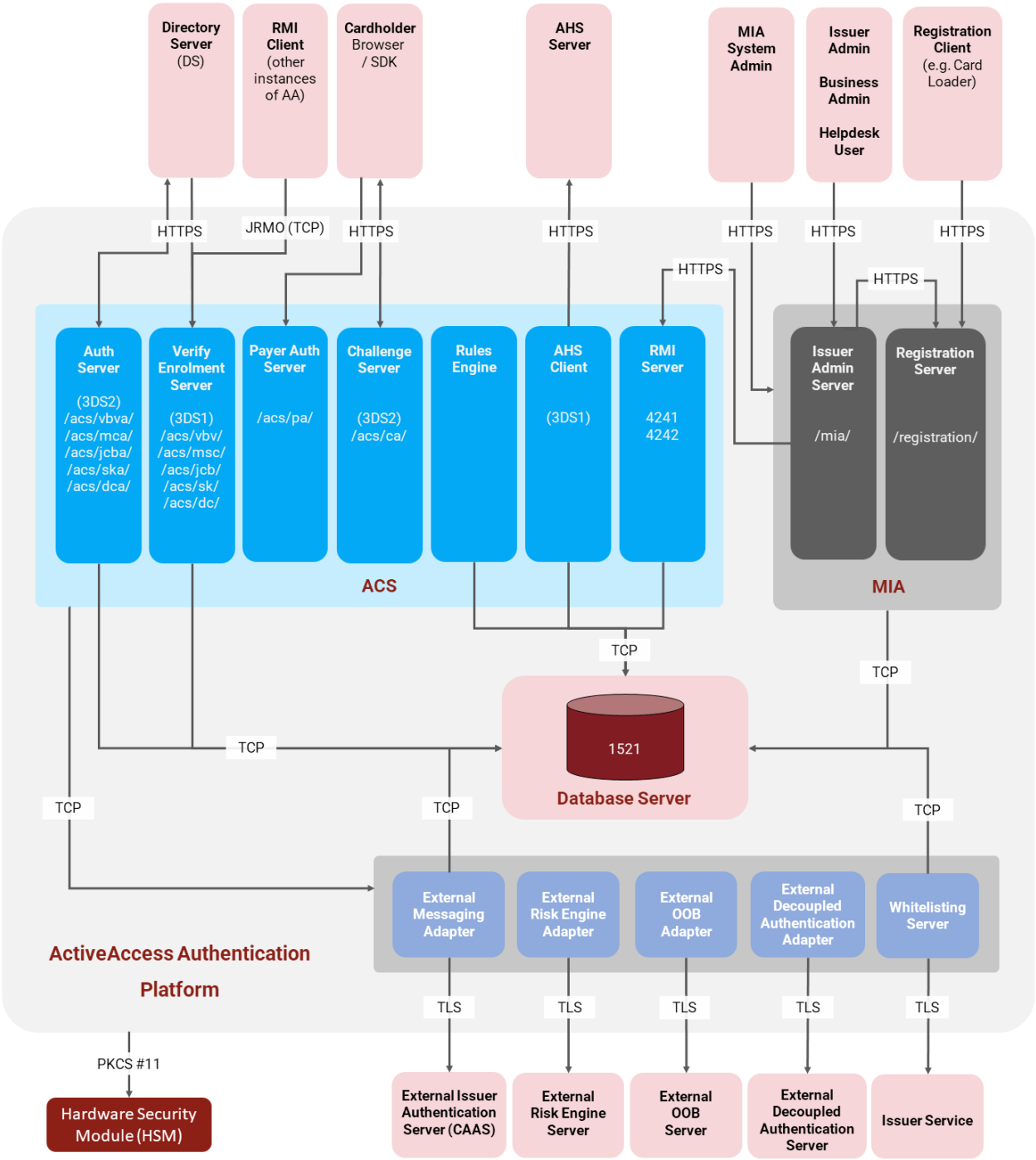
Inbound connections: Authentication server, Verify enrolment server, RMI Server, AHS Client, Rule Engine, External Messaging Adapter, Administration server, Registration server.

Outbound connections: None

Other requirements: None

Logical View of ActiveAccess

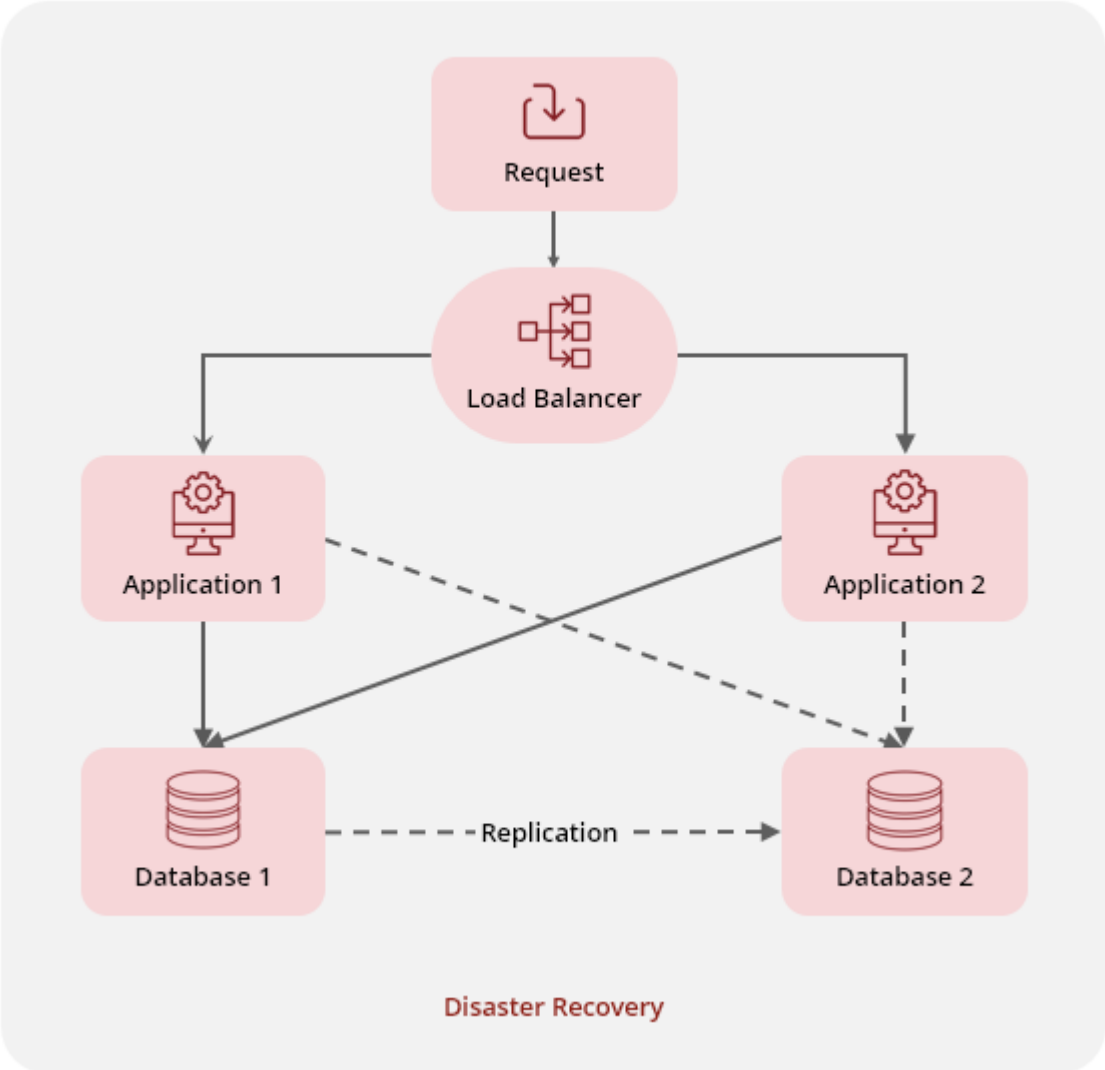
The following diagram displays the logical view of ActiveAccess with the components explained earlier on this page.



Production Setup with Disaster Recovery

In this setup, the ActiveAccess application is setup on Application 1 and Application 2 servers, using one database server (Database 1). Requests sent to the ACS will be forwarded to the Application servers (Application 1 and Application 2), as configured by the load balancer.

Both Application 1 and Application 2 servers will use Database 1. Database 2 is a replication of Database 1, and is on stand-by. If connection to Database 1 fails, Database 2 will be used.

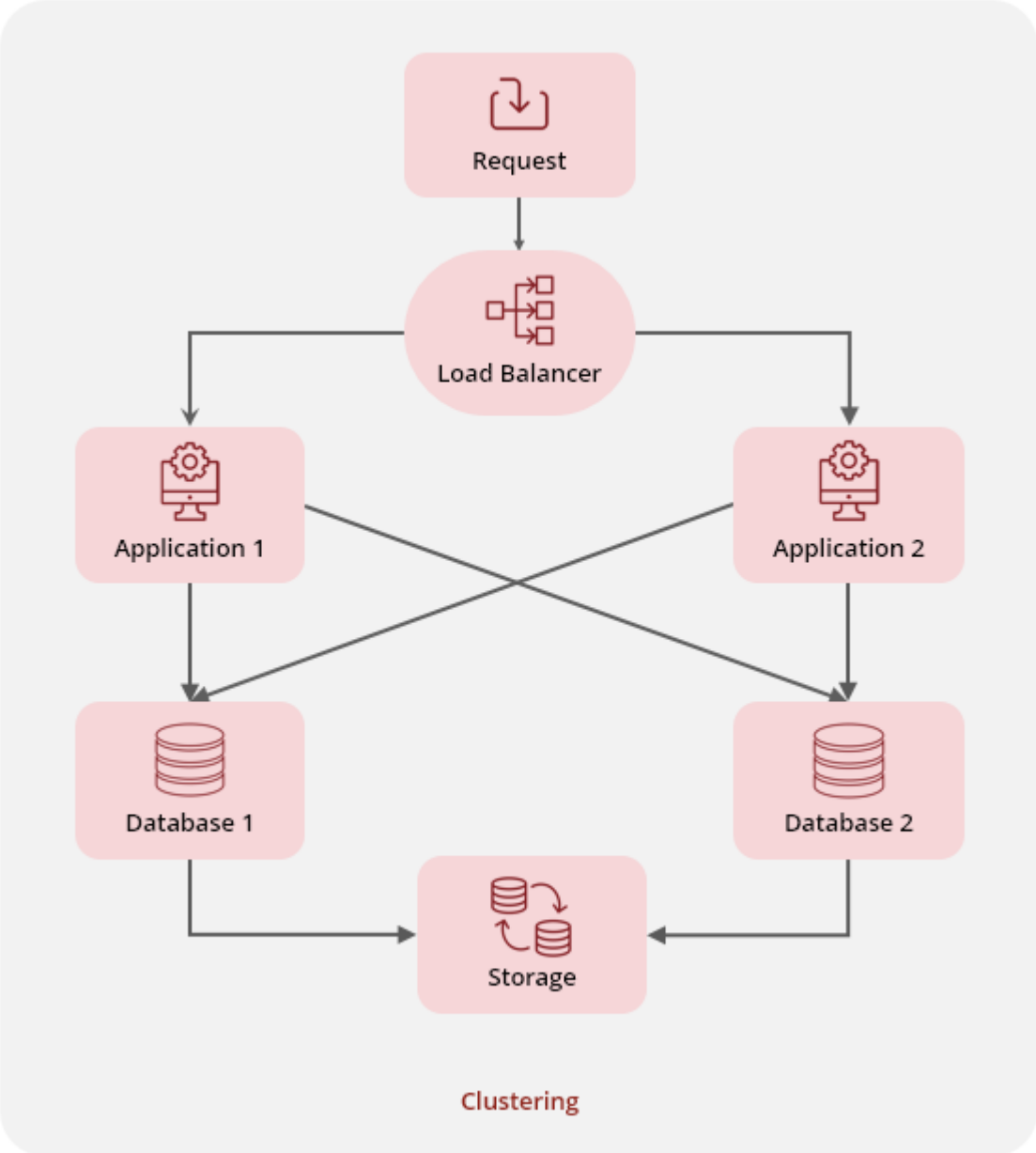


Production Setup with Clustering

In this setup, the ActiveAccess application is setup on Application 1 and Application 2 servers, using two database servers (Database 1 and Database 2) which share the same storage. Requests sent to the ACS will be forwarded to the Application servers (Application 1 and Application 2), as configured by the load balancer.

All application and database servers are active. Application 1 and Application 2 servers will use Database 1 and Database 2 based on the configurations and their ability to establish a connection.

> **Info**
>
> Oracle RAC can be used for the database clustering.



Hardware and Software Requirements

Minimum Hardware Requirements	
Processor	- Intel® Xeon® X5550, or equivalent - 16GB RAM

Minimum Hardware Requirements	
Hardware Security Module (HSM)	- PKCS #11 enabled General Purpose HSMs (with the latest PKCS #11 driver as recommended by the HSM vendor) - Sun JCE (for testing purposes)

Software Requirements	
JDK	- Oracle JDK 1.8 - OpenJDK 1.8
Application Server	- Apache Tomcat 8 - Apache Tomcat 9 - Oracle WebLogic Server 14c (14.1.1.0.0)
Database	- Oracle 11g - Oracle 11gXE - Oracle 12c - Oracle 19c

1. A proprietary wire-level protocol designed by Sun Microsystems to transport Java RMI. JRMP serves the same function as IIOP, but also supports object passing. It is also referred as the "RMI transport protocol" for Java

External Components

Installation of External Components

External Components

Java Development Kit (JDK)

JDK can be freely downloaded from Sun Microsystems at http://java.sun.com/. JDK must be installed with the default settings. Follow the on screen installation instructions for the JDK to complete the installation.

ActiveAccess requires the installation of Oracle JDK 1.8 or OpenJDK 1.8. It is generally advisable that you install the latest minor version within a supported JVM.

You must only use one of the specified JVM versions. This is referred to as a compatible JDK in this document. Note that a newer version of JVM may not necessarily be backward compatible.

Hardware Security Module

ActiveAccess supports PKCS #11 Cryptographic API. For installation of the HSM module, please refer to your HSM manual.

Note

For testing purposes, you can use the Sun JCE provider, available during setup.

Installing the HSM module

- The path of the PKCS #11 library file will need to be specified during ActiveAcces installation.
- The slot number must be selected during ActiveAccess installation.
- The PIN created during the installation of your HSM will be required during ActiveAccess installation.

Thales e-Security HSM

If you are using a Thales e-Security nShield HSM, the environment variable `CKNFAST_OVERRIDE_SECURITY_ASSURANCES` is required to be set for key generation.

LINUX

- Edit the startup file (~/.bashrc)
- Add the following to the end of the file:

```
export CKNFAST_OVERRIDE_SECURITY_ASSURANCES=all
```

- Save and close the file.
- Load the startup file using the following:

```
\$ source ./profile
```

- Verify that the variable is set by executing the following:

```
echo \$CKNFAST_OVERRIDE_SECURITY_ASSURANCES
```

The output should be `all`.

WINDOWS

- In your system's **Control Panel\System and Security\System**, click on **Advanced system settings** link.
- Click **Environment Variables…**.
- In the System variables section, create a new environment variable:

  Variable name: `CKNFAST_OVERRIDE_SECURITY_ASSURANCES`

  Variable value: `all`
- To verify if the variable has been set, open a new Command Prompt window, and execute the following:

```
echo %CKNFAST_OVERRIDE_SECURITY_ASSURANCES%
```

The output should be `all`.

Application Server

ActiveAccess supports Java Application Servers compatible with Servlet specification 3.0. Install your preferred compatible application server with default settings. Please follow the installation instructions from the application server's documentation.

Tomcat

Tomcat is freely available for download from Apache at http://tomcat.apache.org/.

- Install Tomcat with default settings. Please follow Tomcat installation instructions from the Tomcat documentation.
- Tomcat HTTP server starts on port 8080 by default. In order to change the port settings edit **Tomcat/conf/server.xml**
- Update the following section in the configuration for this port number:

```
<!-- ==================== Connectors ==================== -->

<!-- Normal HTTP Connector -->

<Connector executor="tomcatThreadPool"

port="8080" protocol="HTTP/1.1"

connectionTimeout="20000"

redirectPort="8443" />
```

Note

If you are using Tomcat 7, the following `init-param` should be added to the JSP servlet tag (`<servlet-name>jsp</servlet-name>`)

```
<init-param>
  <param-name>compilerSourceVM</param-name>
  <param-value>1.7</param-value>
</init-param>
<init-param>
  <param-name>compilerTargetVM</param-name>
  <param-value>1.7</param-value>
</init-param>
```

Configuring SSL

ActiveAccess requires that communication between client and server uses HTTPS. Configure the application server to run in HTTPS mode.

Tomcat SSL Configuration

To configure Tomcat running in HTTPS mode, please refer to the following:

For Tomcat 8.0+: https://tomcat.apache.org/tomcat-8.0-doc/ssl-howto.html

For Tomcat 8.5+: https://tomcat.apache.org/tomcat-8.5-doc/ssl-howto.html

Please note Tomcat supports two modes of SSL Connectors: JSSE and APR, for which the configuration is different; please refer to the relevant configuration sections in the above Tomcat documentation, for details.

An example configuration for JSSE SSL configuration taken from the Tomcat 8.0 documentation is provided below:

Create KeyStore (using Java Keytool):

- To create a new Java KeyStore from scratch, containing a single self-signed Certificate, execute the following from a terminal command line:

WINDOWS

```
"%JAVA_HOME%\bin\keytool" -genkey -alias appserver -keyalg RSA
```

UNIX

```
\$JAVA_HOME/bin/keytool -genkey -alias appserver -keyalg RSA
```

(The RSA algorithm should be preferred as a secure algorithm, and this also ensures general compatibility with other servers and components.)

This command will create a new file, in the home directory of the user under which you run it, named ".keystore". To specify a different location or filename, add the -keystore parameter, followed by the complete pathname to your KeyStore file, to the keytool command shown above. For example:

WINDOWS

```
"%JAVA_HOME%\bin\keytool" -genkey -alias appserver -keyalg RSA

\-keystore \path\to\my\keystore
```

UNIX

```
\$JAVA_HOME/bin/keytool -genkey -alias appserver -keyalg RSA

\-keystore /path/to/my/keystore
```

You will also need to reflect this new location in the application server's configurations, for example, server.xml configuration file for Tomcat:

Example

Configure the Tomcat connector (in the file **TOMCAT_HOME/conf/server.xml**)

```
<!-- Define a SSL Coyote HTTP/1.1 Connector on port 8443 -->
<Connector
protocol="org.apache.coyote.http11.Http11NioProtocol"
port="8443" maxThreads="200"
scheme="https" secure="true" SSLEnabled="true"
keystoreFile="${user.home}/.keystore" keystorePass="changeit"
clientAuth="false" sslProtocol="TLS"/>
```

Bypassing the HSM Password Dialog Box

ActiveAccess displays a dialog box for HSM password entry, when you start Tomcat.

- In order to suppress the dialog box and enter the password in the console, add the following parameter to JAVA_OPTS in the catalina.sh file of Tomcat:

`\-Dconsole`

- *Or alternatively*, you can directly bypass the HSM password by adding the following line in **activeaccess.properties** configuration file (located in the AA_HOME directory created during installation):

  `HSM_PASSWORD= < password >`

  Replace `< password >` with the base64 encoded format of your HSM password.

Base64 encoding

For base64 encoding, the following command can be used in Ubuntu Linux to generate a base64 encoded string with default settings:

```
echo -n 'YourPlainTextPassword' | base64
```

The output of the command above will be:

```
WW91clBsYWluVGV4dFBhc3N3b3Jk
```

Increasing the Java Heap Size

JRE allocates 64MB of heap memory to a Java process by default. It is quite often necessary to increase this rather conservative memory allocation for server applications.

Tomcat

To increase the heap size available to Tomcat add the following line to catalina.bat (Windows) or catalina.sh (UNIX):

```
set JAVA_OPTS= -Xms<min_heap> -Xmx<max_heap>
```

For example in order to set the minimum heap size to 256MB and allow the heap to grow up to 512MB use:

```
set JAVA_OPTS= -Xms256m –Xmx512m
```

Oracle Database

Character Set

The database character set **must** be AL32UTF8 to support all Unicode characters.

User Name and Password for a database

This is the user name and password that you use to access the database. You may set these database user names to the same user (schema) that you have specified for the database owner (The schema that holds all ActiveAccess database objects). However, if you wish to reserve the database owner for administration purposes and set up a more restricted user for ActiveAccess to access the database schema, please grant the following permissions to the restricted database user:

These permissions require confirmation:

```
Objects: EXECUTE
```

```
PL/SQL: EXECUTE
```

```
Sequences: ALTER, SELECT
```

```
Tables: DELETE, INDEX, INSERT, REFERENCES, SELECT, UPDATE
```

Oracle 19c

If you are using Oracle 19c, add the following privileges for the database user before the installation of ActiveAccess:

```
grant execute on DBMS_SCHEDULER to USERNAME;

grant create job to USERNAME;
```

Note

Please refer to your database server documentation for the installation and configuration of Oracle server.

Configuring DCD (Dead Connection Detection)

Set the optional parameter `SQLNET.EXPIRE_TIME` to 10 (for 10 minutes) in the sqlnet.ora configuration file.

The configuration file is normally located at **$ORACLE_HOME/network/admin** directory.

The value of this parameter determines how often SQL*NET attempts to verify that the connection is still alive. This is to prevent shadow connections to be left open indefinitely.

There are a number of processes that hold a permanent or temporary lock on the database. If the connection to database is abruptly terminated (network disconnected or the server is turned off), the lock remains and will not be reclaimed by other competing processes. This affects sending notification messages via email, scheduling card upload and user upload jobs or registration services.

Configuring DCD ensures that this situation is automatically rectified after the specified time out.

Connection Pooling and Firewall

This section provides important operational information for proper configuration of the environment, when the database server is behind a firewall.

ActiveAccess components use a technique known as **connection pooling** to improve the performance of database related tasks. Connection pooling improves performance by reusing previously established connections. However, this may cause a problem when the database server is behind a firewall. The usual symptom is that the application appears to become unresponsive or frozen after a long period of inactivity.

This is due to firewall idle connection time-out setting. A firewall typically drops idle connections after a configurable time-out has expired. This causes further data transmission through these connections to be ignored by the firewall. Since most firewalls simply ignore the data packets and do not respond, this leaves the sender in a state of wait. The length of this wait state depends on the operating system's time-out setting. For Windows this is typically 15 seconds while the default Solaris time-out is 8 minutes during which the application appears to be frozen.

To prevent this problem ActiveAccess and ActiveIssuer components close idle database connections after 15 minutes. Make sure that your firewall time out setting is at least 1 minute longer than the default application idle connection time out.

The default can be changed by setting the DB_IDLE_TIMEOUT configuration option (in seconds) for each component.

Find Transactions Performance

The performance of transaction search can be greatly improved by analysing the HISTORYSESSIONS table on a regular basis.

- Run the following SQL commands on the database monthly:

```
analyze table HISTORYSESSIONS compute statistics;
analyze table CARD compute statistics for all indexed columns;
analyze table CARDDATA compute statistics for all indexed columns;
analyze table REQUEST compute statistics for all indexed columns;
```

Analysing a table can take a long time and puts extra load on the database. Analyse the tables at a time when database activity is low.

Two-Factor Authentication Devices

SMS

SMS authentication is natively supported by ActiveAccess and does not require additional software. However, ActiveAccess needs to be configured to send SMS messages using SMPP protocol to an SMSC (SMS Centre). ActiveAccess supports SMPP-API-0.3.9.1. An SMSC is normally a gateway to the mobile communication network provided by a Telco or third party service provider.

You need the following details in order to configure SMS authentication in ActiveAccess administration:

Name: A unique name to identify this SMS centre in ActiveAccess

IP: The IP address of the SMS Centre

Port: The port which that SMS Centre is listening on

System ID: The username that is used by SMS Centre for authentication

Password: The password that is used by the SMS Centre for authentication

Sender's mobile number: The mobile number displayed to the message recipient.

> **Note**
>
> Note that to be able to send SMS with templates other than English language or using symbols in SMS Template, you must set following system property in the **TOMCAT_HOME/bin/catalina.bat** or **catalina.sh**:
>
> -Dsmpp.default_alphabet=ie.omk.smpp.util.UCS2Encoding

There are two ways to send OTP to SMSC:

MAILTO

IP: `MailTo:$DEVICE_SERIAL_NUMBER@example.com`

`$DEVICE_SERIAL_NUMBER` will be replaced by ACS with the mobile number that is stored for the card.

> **Note**
>
> To use this option, mail server must be configured in **System Management > Settings**.

SMS VIA JMS

- Approach 1:
  IP: `SmsViaJms:[IP_ADDR_STAND_ALONE_APP]`
- Approach 2:
  IP: `SmsViaJms`

Note

Note that to be able to send SMS with templates other than English language or using symbols in SMS Template, you must set following system property in the **TOMCAT_HOME/bin/catalina.bat** or **catalina.sh**:

-Dsmpp.default_alphabet=ie.omk.smpp.util.UCS2Encoding

Email OTP

Email authentication is natively supported by ActiveAccess and does not require additional software. However, ActiveAccess needs to be configured to send OTP via Email. You need the following details in order to configure Email authentication in ActiveAccess administration:

Mail server address: The address of the mail server

Mail server port: The port which the mail server is listening on

Mail server username: The username that is used by the mail server for authentication

Mail server password: The password that is used by the mail server for authentication

Mail server protocol: The protocol that is used by the mail server for secure communications over the network

Mail sender: The sender's name displayed to the email recipient.

VASCO

To enable authentication using VASCO tokens you need to:

- Install VASCO native libraries first.
- Obtain a copy of Java library 'aal2wrap.java' form VASCO and copy to the lib folder of your ActiveAccess application server.

  The native library should be accessible to the java application. For this purpose in UNIX the variable LD_LIBRARY_PATH should contain the address of the native library which normally is /opt/vasco/VACMAN_Controller-3.4/lib.

  In Windows the address of the DLL file should be added to the PATH variable. Also the VASCO token keys should be uploaded in the system. These files are provided by VASCO with the devices and can be uploaded to ActiveAccess using the administration interface.
- Browse to **System Management > Device Management** choose **upload file** and then specify the location of the file and relevant parameters.

OOB

To enable authentication using OOB, register and configure adapters. For more information, refer to Out of Band Authentication Adapter.

Decoupled Authentication

To enable authentication using Decoupled Authentication, register and configure adapters. For more information, refer to Decoupled Authentication Adapter.

Installation

Prerequisites

- Ensure that a compatible JDK is installed
- Ensure that the hardware security module is properly installed and configured

> **HSM keys**
>
> If this is a first time installation, ActiveAccess keys will be generated automatically.
>
> For subsequent installations of ActiveAccess on other servers ensure that the AES (128 Bits) key alias `AA_MASTER` has been transferred from the primary installation in the current instance of HSM used by the ActiveAccess which is being installed.

- Ensure that the application server is properly installed and configured
- Ensure that the database server is properly installed and you have created a database for ActiveAccess.

> **Database details**
>
> Have the database name, username and password and address at hand for the installation process.

Pre Installation Configurations

Upgrades

For upgrades from **any** version of ActiveAccess to the latest version of ActiveAccess, follow the steps below.

Before the upgrade:

1. Shutdown all instances of ActiveAccess, stop the current Tomcat servers.
2. Back up ActiveAccess directories, including the application server directories and configuration directories, such as AA_HOME. Archive the ActiveAccess directory and store in a safe place. Do this for all instances of ActiveAccess.
3. Back up the database. The upgrade contains schema level changes. You will not be able to roll back, unless the database is fully backed up.
4. Back up all the HSM key data.

- If you have previously deployed **enrolment.war** to your application server, you must remove it.

  For example, for Tomcat, go to **TOMCAT_HOME/webapps**, and remove the `enrolment.war` file and the deployed `enrolment` directory.

- Go to **TOMCAT_HOME/lib**. If the following files exist, back up and remove them:
  - gpcomp.pki-*.jar
  - gpcomp.hsm-*.jar
  - spp-*.jar
  - nfjava-*.jar
  - lunaprovider-*.jar
  - kmjava-*.jar
  - kmcsp-*.jar
  - jprov-*.jar
  - commons-codec-*.jar
  - aal2wrap-*.jar
- It is recommended to replace the HSM-related JAR files provided by ActiveAccess, with the libraries provided by your HSM provider. Note that the name of the replacement JAR files must be exactly the same as the name of the JAR files in the ActiveAccess installation package. For example, the `jprov.jar` file that is provided by your HSM provider may need to be renamed to `jprov-1.1.jar`.

Upgrades to v8.5.x and later

For upgrades to **ActiveAccess 8.5.x and later**, all clients must have **PKCS #11** configured for connectivity to the HSM (this excluses ActiveAccess installations with SunJCE).

- If your ActiveAccess installation already uses PKCS #11 (`HSMPROVIDER=PKCS11`), no changes are required. This would be the case if the first version of ActiveAccess that you installed was 7.4.x or later.
- If your ActiveAccess installation does not utilise PKCS #11 (i.e. the first version of ActiveAccess that you installed was version 7.3.x or older, with `HSMPROVIDER=ERACOM` , `HSMPROVIDER=nCipherKM` , or `HSMPROVIDER=LunaProvider`), you must add the following attributes in **activeaccess.properties** and set an appropriate value for them:
  - `MASTER_HSM_LIB_DIR=`
  - `MASTER_HSM_SLOT=`
  - `PKCS11_CONFIG_FILE_PATH=`

> **Note**
>
> If you are migrating to a new HSM device, the values set for the attributes `MASTER_HSM_LIB_DIR` , `MASTER_HSM_SLOT` , and `PKCS11_CONFIG_FILE_PATH` must be for the new HSM device.

Upgrades from v7.x.x to v8.x.x and later

If you are upgrading from ActiveAccess v7.x.x, **in addition** to the upgrade steps above, follow the steps below.

- An `AA_HOME` directory is required from which ActiveAccess will load the configurations it requires for installation. Create a directory and set an `AA_HOME` environment variable to this directory.

> **Note**
>
> Refer to your Operating System and application server documentation for any specific instructions for setting an environment variable.
>
> - AA_HOME can be set in Tomcat in catalina.bat/catalina.sh as JAVA_OPTS
> - AA_HOME can be set in WebLogic in setDomainEnv.cmd or startWebLogic.sh

- Add the following line in **AA_HOME/activeaccess.properties**

`HSM_PASSWORD= < password >`

Replace `< password >` with the base64 encoded format of your HSM password.

> **Warning**
>
> After the installation, a new configuration file, `activeaccess.properties`, will be created automatically in the `AA_HOME` directory. This new configuration file combines `acsconfig.properties`, `miaconfig.properties` and `regconfig.properties` and these files will be removed during the installation process.
>
> If you have configured any parameters that are not specific to ActiveAccess, you must take a back up of these files before running the installation and move these parameters manually to `activeaccess.properties`.

New installations

- An `AA_HOME` directory is required from which ActiveAccess will load the configurations it requires for installation. Create a directory and set an `AA_HOME` environment variable to this directory.

> **Note**
>
> Refer to your Operating System and application server documentation for any specific instructions for setting an environment variable.
>
> - AA_HOME can be set in Tomcat in catalina.bat/catalina.sh as JAVA_OPTS
> - AA_HOME can be set in WebLogic in setDomainEnv.cmd or startWebLogic.sh

- In the installation package, go to the **ActiveAccess** directory, copy **activeaccess.properties** to your `AA_HOME` directory.
- Open **activeaccess.properties** and fill in the required configuration parameters.
- ✨ It is recommended to replace the HSM-related JAR files provided by ActiveAccess, with the libraries provided by your HSM provider. Note that the name of the replacement JAR files must be exactly the same as the name of the JAR files in the ActiveAccess installation package. For example, the `jprov.jar` file may need to be renamed to `jprov-1.1.jar`.

Deploying WAR Packages

Download and extract the ActiveAccess installation package from **GPayments MyAccount > ActiveAccess > Download**.

Access Control Server, Administration Server, Registration Server and Whitelist Server are distributed in the ActiveAccess installation package as WAR packages. To install these packages, deploy **acs.war**, **mia.war**, **registration.war** and **whitelist.war** packages from **ActiveAccess/files** to your application server.

Deployment mechanism

Depending on the application server, the deployment mechanism would be different.

For example:

For Tomcat, the war files should be copied to **TOMCAT_HOME/webapps**.

For Oracle WebLogic Server, extract .war files and use the extracted directory to copy them in autoDeploy directory, or use the extracted directory in WebLogic's manual deployment (WebLogic console > domainStructure > Deployments > install section).

Please refer to your application server's documentation for instructions.

Installation

To initialize the installation process, start the application server.

This process may take a couple of minutes to complete.

An installation log will be created in **AA_HOME/logs/install_log.log**.

Info

If you are using two different database users in setup (for `db_owner` and `db_user`), from ActiveAccess v8.0.1 onwards, grant scripts are run automatically during setup and no longer need to be run manually.

> **Warning**
>
> ActiveAccess modules have specific configuration files such as log4j.xml, sms_jms_config.properties, which allow the client to customise various parameters based on their environment settings.
>
> In some releases, new parameters are introduced or deprecated. The installer will compare the dates of the configuration files in the installation package with the ActiveAccess working directory and raise warnings if there are any differences.
>
> Following each update/upgrade, the **install_log.log** file should be checked by the Admin for warnings in order to ensure that no changes in the configuration files have been missed.
>
> The warnings will appear in the following format:
>
> ```
> The date or size of [full path of the config file in installation package]
> is different from [full path of the config file in AA_HOME], compare the
> content and make sure all the required and optional parameters are OK.
> ```

Installation of Individual Components

The Access Control Server handles greater loads than other components and may be installed on a physical machine, dedicated to transaction processing.

Administration, Registration and ➕ Whitelist servers are usually installed on the same physical machine.

To install individual components:

- Ensure that you have the prerequisites properly installed and configured for each component that is being installed individually.
- Deploy the component's WAR package to the application server.
  - Access Control Server: **acs.war**
  - Administration Server: **mia.war**
  - Registration Server: **registration.war**
  - ➕ Whitelist Server: **whitelist.war**
- Configure the **installation parameters** (**AA_HOME** directory and configuration file).
- Start the application server.
- Ensure that the AES (128 Bits) key alias `AA_MASTER` exists in the HSM.

> **Tip**
>
> If this is a first time installation, ActiveAccess keys will be generated automatically.
>
> For subsequent installations of ActiveAccess on other servers ensure that the AES (128 Bits) key alias `AA_MASTER` has been transferred from the primary installation in the current instance of HSM used by the ActiveAccess which is being installed.

Rollback Process

In case you need to roll back to the previous version, follow the steps below:

1. Shutdown all ActiveAccess servers and stop the applications in the application server.
2. Restore the original database.
3. Restore ActiveAccess directories and deploy the previous version of the applications on your application server locations.

Post Installation

- If you upgraded to ActiveAccess v9.0.x from an older version, you must run the Migrate to Data Key Utility and ensure the key migration process completes successfully.

> **Note**
>
> This utility does not need to be run for new installations of ActiveAccess v9.0.x.

> **Warning**
>
> It is strongly recommended to start the key migration process as soon as possible. Running this utility is critical to upgrading to future versions of ActiveAccess. Old keys (RSA, CAVV, AAEV, HMAC, HMAC256, encryption keys) and old Issuer Signing Certificates will not be supported in future releases of ActiveAccess beyond January 2022.
>
> If you have any questions in regards to the above, please contact techsupport@gpayments.com.

- On successful installation and when the application server is started, the internal components are started on the default port. These components are:

Access Control Server

Base URL: `https://< server-address >:< port >/acs/`

The following extensions can be added to the base URL:

Card Scheme	3DS1 VE/UE	3DS1 PA/UA	3DS2 AReq	3DS2 CReq
Verified by Visa/Visa Secure	/vbv	/pa	/vbva	/ca
Mastercard SecureCode/IDC	/msc	/pa	/mca	/ca
JCB J/Secure	/jcb	/pa	/jcba	/ca
American Express SafeKey	/sk	/pa	/ska	/ca
Diners Club International ProtectBuy	/dc	/pa	/dca	/ca

Example

Verified by Visa VE: `https://< server-address >:< port >/acs/vbv`

Info

The PA and CReq paths determine the **ACS URL** as seen by the user.

3DS Method URL: `https://< server-address >:< port >/acs/tdsmethod`

Monitoring the availability of ACS: `https://< server-address >:< port >/acs/ping`

Info

If the ACS is up and running, a JSON message will be displayed, which reports the availability of Database as well as the HSM. If the ACS is down, an error will be displayed. If Database or HSM is unavailable the value will be “not connected” in displayed message.

JSON Response Elements:

Attribute	Possible value
dbConnectionStatus	- Connected - Connection limit reached - Can't establish connection - Connection pool is not initialized
hsmConnectionStatus	- Connected - Not connected

Example

{"dbConnectionStatus":"connected","hsmConnectionStatus":"connected"}

Administration Server

Base URL: `https://< server-address >:< port >/mia/`

Monitoring the availability of MIA: `https://< server-address >:< port >/mia/ping`

Info

If the Administration Server is up and running, a JSON message will be displayed, which reports the availability of the Database as well as the HSM. If the Administration Server is down, an error will be displayed. If the Database or HSM is unavailable the value “not connected” will be displayed in the message.

JSON Response Elements:

Attribute	Possible value
dbConnectionStatus	- Connected - Connection limit reached - Can't establish connection - Connection pool is not initialized
hsmConnectionStatus	- Connected - Not connected

Example

{"dbConnectionStatus":"connected","hsmConnectionStatus":"connected"}

Registration Server

Base URL: `http(s)://< server-address >:< port >/registration/`

Info

Entering the URL above in a browser will display the message:

```
The Registration Server has received a GET.

Your signed XML (application/xml) should be sent via HTTP POST.
```

Login to the Administration Server as Administrator and set the **Registration server URL** in the **System Management/Settings** section to the base URL of the Registration server.

The Registration Server accepts HTTP Post commands for the purpose of uploading cardholder registration data.

Info

When using SSL, the Registration server certificate should be signed by a public CA. If you intend to use a self-signed certificate or a certificate signed by a certificate authority other than commercially known certificate authorities, you must import the CA's root certificate into the Administration server's TrustStore.

The Administration server TrustStore (cacerts) can be found in the config directory of the Administration server. Export your CA root certificate as a DER encoded or Base-64 encoded X509 certificate and use Keytool to import this into the cacerts file:

```
keytool -import -trustcacerts -alias myca -file cacert.cer -keystore cacerts -storepass changeit
```

Replace cacert.cer with the CA certificate file you wish to add to the KeyStore.

The following extensions can be added to the base URL:

Process	URL Extension
Card registration requests	/card
User registration requests	/user
Notification report requests	/notification

Note

The base URL can be used for **card registration requests**. Using the extension is optional.

Monitoring the availability of Registration: `http(s)://< server-address >:< port >/registration/ping`

Info

If the Registration Server is up and running, a JSON message will be displayed, which reports the availability of Database as well as the HSM. If the Registration Server is down, an error will be displayed. If Database or HSM is unavailable the value will be “not connected” in displayed message.

JSON Response Elements:

Attribute	Possible value
dbConnectionStatus	- Connected - Connection limit reached - Can't establish connection - Connection pool is not initialized
hsmConnectionStatus	- Connected - Not connected

Example

{"dbConnectionStatus":"connected","hsmConnectionStatus":"connected"}

Whitelist Server

Base URL: `http(s)://< server-address >:< port >/whitelisting/wl/api/merchant/`

Login to the Administration Server as an **Administrator** user and set the **Whitelist server URL** in **System Management/Settings** to the base URL of the Whitelist server. The Whitelist Server accepts HTTP Post commands for the purpose of adding/displaying/removing cardholder’s whitelisted merchant data.

Info

When using SSL, the Whitelist server certificate should be signed by a public CA. If you intend to use a self-signed certificate or a certificate signed by a Certificate Authority other than commercially known certificate authorities, you must import the CA's root certificate into the Administration server's TrustStore. The Administration server TrustStore (cacerts) can be found in the config directory of the Administration server. Export your CA root certificate as a DER encoded or Base-64 encoded X509 certificate and use Keytool to import this into the **cacerts** file:

```
keytool -import -trustcacerts -alias myca -file cacert.cer -keystore cacerts -storepass changeit
```

Replace **cacert.cer** with the CA certificate file you wish to add to the KeyStore.

The following extensions can be added to the base URL:

Process	URL Extension
Add Merchant request	/add
Remove Merchant request	/remove
Display Merchant request	/getMerchant
Remove list of Merchant request	/removelist
Display history of Merchant request	/getMerchantHistory

Configuration Files

ActiveAccess Configuration File

AA_HOME/activeaccess.properties

The ActiveAccess Configuration file, **activeaccess.properties**, is automatically created/updated by the ActiveAccess installation. Common options such as database information are required to be configured during installation. The following sections document all the available parameters in case you need to change the defaults.

> **Note**
>
> ActiveAccess server must be restarted for changes to configuration files to take effect.

Common Configuration Parameters

DBNAME, DBOWNERPASSWORD

This is the database owner name and password that you use to create the database. When you first set or change the database owner password, you may set it in clear text. You should also add (PLAIN_TEXT=) to your configuration file.

> **Note**
>
> This parameter must always have a value.

DBUSERNAME, DBPASSWORD

This is the **username** and **password** that you use to access the database. In a simple configuration this username may be the same as the database owner name. When you first set or change the database password, you may set it in clear text. You should also add (PLAIN_TEXT=) to your configuration file.

> **Note**
>
> This parameter must always have a value.

PLAIN_TEXT=

This instructs the server to read DBOWNERPASSWORD and DBPASSWORD in clear text and replace them with the encrypted values.

DBURL

For Oracle the default URL is:

```
jdbc\:oracle\:thin\:\@127.0.0.1\:1521\:ORCL
```

Replace 127.0.0.1:1521 with the IP address and port number of the Oracle instance you have installed. ORCL is the SID of the database and must be replaced with the SID you selected during the installation of the database server.

```
DBURL=jdbc\:oracle\:thin\:\@192.168.0.202\:1521\:ORCL
```

DBDRIVER

For Oracle, leave the default value unchanged as shown below:

```
DBDRIVER=oracle.jdbc.driver.OracleDriver
```

INITIALCONNECTIONS

Specifies the initial length of database connection pool allocated by the application.

MAXCONNECTIONS

Specifies the maximum length of database connection pool that can be allocated by the application.

WAITIFBUSY

Can be set to either true or false. The default is true. When set to true, connection requests exceeding the maximum connections will be queue until a connection is freed. When set to false, the application immediately returns an connection erorr if no free connection can be found in the pool.

DB_IDLE_TIMEOUT

The database idle connection time out in seconds. Idle database connections are closed in the application's connection pool after the specified time. The default is 900 seconds.

DBENCODED

If this parameter sets to false reading and writing to database is done in ISO-8859-1 character set and ActiveAccess uses its own encoding (Default value is **false**). Otherwise database's own encoding is used.

HSMPROVIDER

Used to specify the HSM provider name.

For ActiveAccess instances which were originally installed prior to ActiveAccess v7.4.0, the value would be **nCipherKM** for Thales e-Security, **ERACOM** for SafeNet, or **SUN** for Sun JCE. In ActiveAccess instances originally installed after and including v7.4.0, this parameter would be **PKCS11** or **SUN**.

> **Note**
>
> This parameter should always have a value.

KEYSTORE_DIR

Used to specify the physical location of the HSM KeyStore (Thales e-Security or SunJCE). Use forward slash as the path separator e.g.: `KEYSTORE_DIR=c:/nfast/kmdata/local`

PKCS11_CONFIG_FILE_PATH

Used to specify the path to the PKCS #11 configuration file with a `.properties` extension.

The contents of the configuration file should contain `library`, `slot` and `name` parameters.

> **Note**
>
> If this file does not exist, it will be generated automatically.

nShieldHSM

Only if you are using an nShield HSM, set the value to Yes. For all other HSM types, it should be left blank.

HSM_PASSWORD

Used to set the HSM password in the configuration file. This option takes precedence over the java option `-Dcom.gpayments.hsm.password`. The HSM password must be provided in base64 encoded format in both cases. Leave empty for a blank HSM password.

HSM_LIB_DIR

Used to specify the path of **.dll** or **.so** file which will be added to **pkcs11config.properties** file, if the file does not exist.

HSM_SLOT

Used to specify the slot number that will be added to **pkcs11config.properties** file, if the file does not exist.

MASTER_HSM_LIB_DIR

Used to specify the path of **.dll** or **.so** file which will be added to **pkcs11config.properties** file, if the file does not exist. This will be used for saving the Master Key in the HSM.

> **Note**
>
> This parameter is used for migration to HSM connectivity via PKCS #11.

MASTER_HSM_SLOT

Used to specify the slot number that will be added to **pkcs11config.properties** file, if the file does not exist. This will be used for saving the Master Key in the HSM.

> **Note**
>
> This parameter is used for migration to HSM connectivity via PKCS #11.

HSMENCALIAS

When the MIA/ACS Settings Encryption Key is automatically or manually retired and replaced with a new one using the PCIDSS Key Retiring Utility, the default key alias is changed. Therefore, the new key alias is specified by HSMENCALIAS.

WS_POOL

Used to specify the size of WebSocket pool. The default value is 1000.

TOMCAT_KEYSTORE

Used to specify the path of the Tomcat KeyStore in case the timeout error fails with SSL Handshake in browser-based authentication.

> **Note**
>
> Use forward slash as the path separator.

TOMCAT_KEYSTORE_PASS

Used to specify the password of the Tomcat KeyStore in case `TOMCAT_KEYSTORE` is set.

TOMCAT_TRUSTSTORE

Used to specify the path of the Tomcat TrustStore in case the timeout error fails with SSL Handshake in browser-based authentication and the SSL connection is not one-sided.

> **Note**
>
> Use forward slash as the path separator.

TOMCAT_TRUSTSTORE_PASS

Used to specify the password of the Tomcat TrustStore in case `TOMCAT_TrustStore` is set.

CARD_MOD_10_CHECK

Used to enable/disable mod 10 check when creating cards via the administration interface, for testing purposes. It can be set to `true` or `false`. The default value is `true`.

TESTING_MODE

Can be set to either `true` or `false`. Set it to true during certification testing. Default value is `false`.

PROVIDER_TEST

Can be set to either `true` or `false`. Set it `true` during certification test only if the test card bin is not supported in default providers.xml file. If set true providers_test.xml should be created and placed at `AA_HOME`.

TEST_AUTH_SERVER

Set URL of authentication server. This parameter is developed to support UL tests.

ACS_REFERENCE_NUMBER_TEST

Set ACS reference number during certification test.

TIMEZONE_ID

Used to set the time zone of the application.

Refer to **ActiveAccess/timezones.txt** which has a list of acceptable time zones.

Example

TIMEZONE_ID=Australia/Sydney

Note

This parameter should always have a value.

AMOUNT_FORMATTER

Used to set the reference for the transaction amount format in SMS, email, and authentication pages. The default is **STANDARD**.

Values:

- **STANDARD**: The US-English standard, which includes a comma as the thousands separator and a dot as the exponent separator.
- **LOCALISED**: The local value derived from the installed server. In case the `user.language.format` and `user_country_format` are not set, the default local of the system will be used.

IGNORE_DTD_ORDER_3DS1

Used to enable/disable the checking of the order of the elements provided in 3DS1 requests. The default is **false**.

Values:

- **true**: The order of the elements will not be checked.
- **false**: The order of the elements will be checked.

PURCHASE_DATE_ACCEPT_BALANCE

Used to configure the balance of the purchase date validation. The validation is disabled by default, unless configured.

- **Accepted range**: 60 to 1440 (1 hour to 1 day in minutes).

> **Note**
>
> This is a temporary parameter. The System admin or Issuer admin should configure this option on MIA.

Additional Administration Server Configuration Parameters

UPLOADCACHE_DIR

Used to specify a location to copy uploaded file that VASCO tokens fetched from it. Use forward slash as a path separator e.g.: UPLOADCACHE_DIR=c:/tempdir

MAX_WARNINGS

Specifies the maximum number of warning messages that the administration server will generate while processing VASCO token files before an error is returned. In other words, if processing a VASCO file creates more warnings than this value, the server will terminate processing of the file and will return an error response. If this parameter is not specified, a default value of 50 is used.

Additional ACS Configuration Parameters

COMPUTERNAME

This is the computer name where the ACS is installed.

DOMAINNAME

This is the domain name where the ACS is installed. It must be resolved to an IP address and you must add this host name to `/etc/hosts` or in Windows `C:\WINDOWS\system32\drivers\etc\hosts` before installation.

BINDING_IP_ADDRESS

Used to define the binding IP address of ActiveAccess.

RMI_PORT

The RMI port of ActiveAccess. The default value for the RMI port is 4242. If you decide to change the RMI port, you can edit this value at any time.

AHS_FLAG

Used to enable/disable Authentication History Server. It can be set to either true or false. The default value is true.

CACHING

This option specifies the caching mode for resources. The default is **everyvisit**.

DBENCODED

Can have two values **Yes** or **No**. If your Database is set to use encoding, set this option to **Yes**.

ZLIBOFF

It can be set to either **true** or **false**. When it is set to true, ACS does not inflate ZIP objects. The default value is false.

> **Warning**
>
> This option is for test purposes only. Setting the options to **true** in production will cause interoperability problems with other 3-D Secure components.

Additional Registration Server Configuration Parameters

VERIFICATION

Can be set to either **true** or **false**. When the verification is true, the registration server checks the authenticity of XML messages by validating the XML signature. Disabling verification should be avoided in a production system for security reasons.

REQUEST_LOGGING

Can be set to either **true** or **false**. Used to collect request debug information, intended for testing purposes. This option should not be enabled in production environment.

MAX_WARNINGS

Specifies the maximum number of warning messages that the registration server will generate, before an error is returned. In other words, if a message sent to the registration server creates more warnings than this value, the server will terminate processing the message and will return an error response. If this parameter is not specified the default value of 50 is used.

Notification Report Collector Job Parameters:

Notification Reports are provided based on collected report files by the Notification Report Collector Job on the Registration server. In order to configure this job to collect the required data and cache report files, the following parameters must be set in activeaccess.properties:

LAST_REPORT_TIME

The last time that the notification report collector job was run

Format: DD/MM/YYYY

OFFICIAL_START_HOUR (Deprecated and is no longer used)

The hour that is used as the start hour of the day. Reports are collected based on this hour. Values: 00..23 (default: 00)

OPTOUT_MODE

The flag that specifies whether report collector should collect the last cardholder opt out only or all opt outs.

Values: LAST/ALL (default: ALL)

SCHEDULER_START_TIME

The time that the report collector job starts to collect reports based on LAST_REPORT_DATE

Format: HH:mm:ss GMT(+0:00) (default: -1 to disable job).

Example: Assume LAST_REPORT_TIME=02/02/2009, SCHEDULER_START_TIME=22:30:30, if today is 05/02/2009, report collector starts at 22:30:30 GMT(+0:00) and collects reports from 02/02/2009 00:00 to 05/02/2009 00:00

> **Note**
>
> If SCHEDULER_START_TIME is set to a time in past, the job will be scheduled for tomorrow at the specified time.

NOTIFICATION_FILE_PATH

The path on the server which the report collector job will cache for the collected report files

The default path is a **NotificationReport** directory, located in the deployed directory of Registration on your application server.

NOTIFICATION_REPORT_LIFETIME

The life time of cached report files on the server in DAY. As soon as the report collector job starts, it removes files if their life time period has already passed

Default: -1 to disable

NOTIFICATION_REPORT_REGEN_ISSUERIDS

A comma separated list of the IDs of the issuers that have retired their encryption key using PCIDSS Retiring Utility. As the result of retiring the encryption key of an issuer, the pre-collected notification report files are no longer valid. This list is automatically populated at the end of the utility process to indicate that notification reports should be re-collected for the specified issuers at the next run of the notification report collector job.

Example

NOTIFICATION_REPORT_REGEN_ISSUERIDS= 284357534937385611, 974922143261996848

Providers File

ActiveAccess requires the default card ranges of all providers in order to process incoming 3D-Secure authentication requests. As card schemes may add new card ranges at any time, the providers file allows for the additions to be made manually, when required. The following options can be updated in **providers.xml** under the **AA_HOME** directory.

- **Provider name, provider index, cname and provider ID:** within the < providerInfo > element for each of the providers, there are tags for the provider's name (< providerName >), index (< providerIndex >), card scheme authentication method (< cName >), and provider ID (< providerId >). The following table shows the possible values for the aforementioned tags.

providerName	providerIndex	cName	providerId
Visa	1	vbv	2
Mastercard	2	msc	1
JCB	3	jcb	3
AMEX	4	sk	5

providerName	providerIndex	cName	providerId
DinersClub	5	dc	6

- **Card Range:** the card ranges for each provider are included in the providers file, in the form of minimum range and maximum range. The minimum range should always be lower than', or equal to, the maximum range, with an equal number of digits. You can add any card range to the providers file inside the tag, by copying the tag and inserting the new minimum and maximum ranges. Make sure the newly added card ranges do not overlap with another provider's card ranges. Furthermore, the tag indicates the required number of digits for card numbers, which fall within the specified card range.

Note

If you want to update the providers file, make sure the xml format is followed closely, as any formatting issues may result in ActiveAccess failing to start.

Note

Changes made to the providers file will not take effect immediately, unless the ActiveAccess server is restarted.

About the Issuer Administration Server

The issuer administration server allows multiple issuers and the system operator to share the same infrastructure and application while maintaining a completely separate view of the system. It enables issuers and the system operator to configure the system for their own purposes independently.

Access Levels

User access levels are controlled by assigning, one of six pre-defined, roles to the user. The user role determines which menu items and functions are accessible by the user.

A **read only** option is available, for all user roles, for example, for users in support roles that are not required to add records, edit details or upload files.

The access levels are:

System administrator - the highest level of access in the system with access to system management, issuer management, user management, cardholder management, transactions, reporting and audit logs.

Issuer administrator - provides access to member bank configuration options, cardholder management, transactions, reporting and audit logs for one, or a group of issuers.

IT security - provides dedicated access to audit logs, for one or a group of issuers.

Member administrator - provides dedicated access to the Admins section (administration user management), for one issuer or an issuer group.

Business administrator - business level of access to the system provides access to cardholder management, transactions, reporting and audit logs, for one issuer or an issuer group.

Helpdesk - provides cardholder management and transactions, for one issuer or an issuer group for helpdesk users.

Logging In and Logging Out

Login

To login to the ActiveAccess administration interface you must be previously registered as an administrative user and know your Username and Password. You must also have access to the required one-time passcode, if two-factor authentication is enabled for your user account.

- From your Web Browser make a connection with the Intranet and access the ActiveAccess login page.

  The ActiveAccess administration **Login** screen is displayed.

> **Warning**
>
> If you have forgotten your password, contact your system administrator.
>
> If security has been compromised (such as when you suspect another person has logged in using your username and password) you can login and then change your password using the **Edit Profile** link situated on the top banner.

> **Note**
>
> If you experience login issues after an upgrade, clear your browser's cookies and try again.

- Enter your **Username** and **Password**.

> **Info**
>
> Both **Username** and **Password** are case sensitive.

- Click the **Login** button.
- ActiveAccess supports two-factor authentication for logging into the Administration UI. By default, users are not forced to use two-factor authentication, unless this feature has been enabled during user creation or has been set up by the user in Edit Profile.

> **Note**
>
> To enable this feature, **email notification messages** must be enabled and configured in Settings.

If two-factor authentication login is enabled for your user account, enter your one-time **Passcode**.

> **Google Authenticator for two-factor authentication login**
>
> To use this feature, you must have Google Authenticator installed on a mobile device and have the provided QR code scanned on the app.
>
> If a System Administrator enables this feature for a user, the QR code will be sent to the user's email address. If a user enables this function for their own account, the QR code will be displayed when enabling the feature.
>
> Refer to Install Google Authenticator for setup instructions of Google Authenticator.

- Click the **Login** button.

Upon entering your username and password (and passcode, if required) successfully, you are verified and the first admin page will be displayed. The page that you see will depend on the access rights assigned to your username (**system administration**, **issuer administration**, **business administration**, **IT security**, **member administrator** or **helpdesk**).

> **Note**
>
> If the user logging in does not belong to an Issuer or Issuer Group with a valid license installed, they will not be able to access any administration pages and will be shown the following message:
>
> *The user '**username**' does not belong to an **Issuer** or **Issuer Group** with a valid **3-D Secure enabled** or **Device enabled license** installed.*
> *Please contact your System Administrator.*

Logout

When you have finished using ActiveAccess administration, it is important that you logout from your account, to prevent other users from performing tasks with your username and access level privileges. The Logout function is accessed via the ***Logout*** link displayed on the right of the title bar area.

> **Warning**
>
> It is also important that you logout while leaving your PC unattended.

- Click the *Logout* link.

The Administration **Login** screen is displayed.

- You may now close your browser window.

Issuer Administration Environment

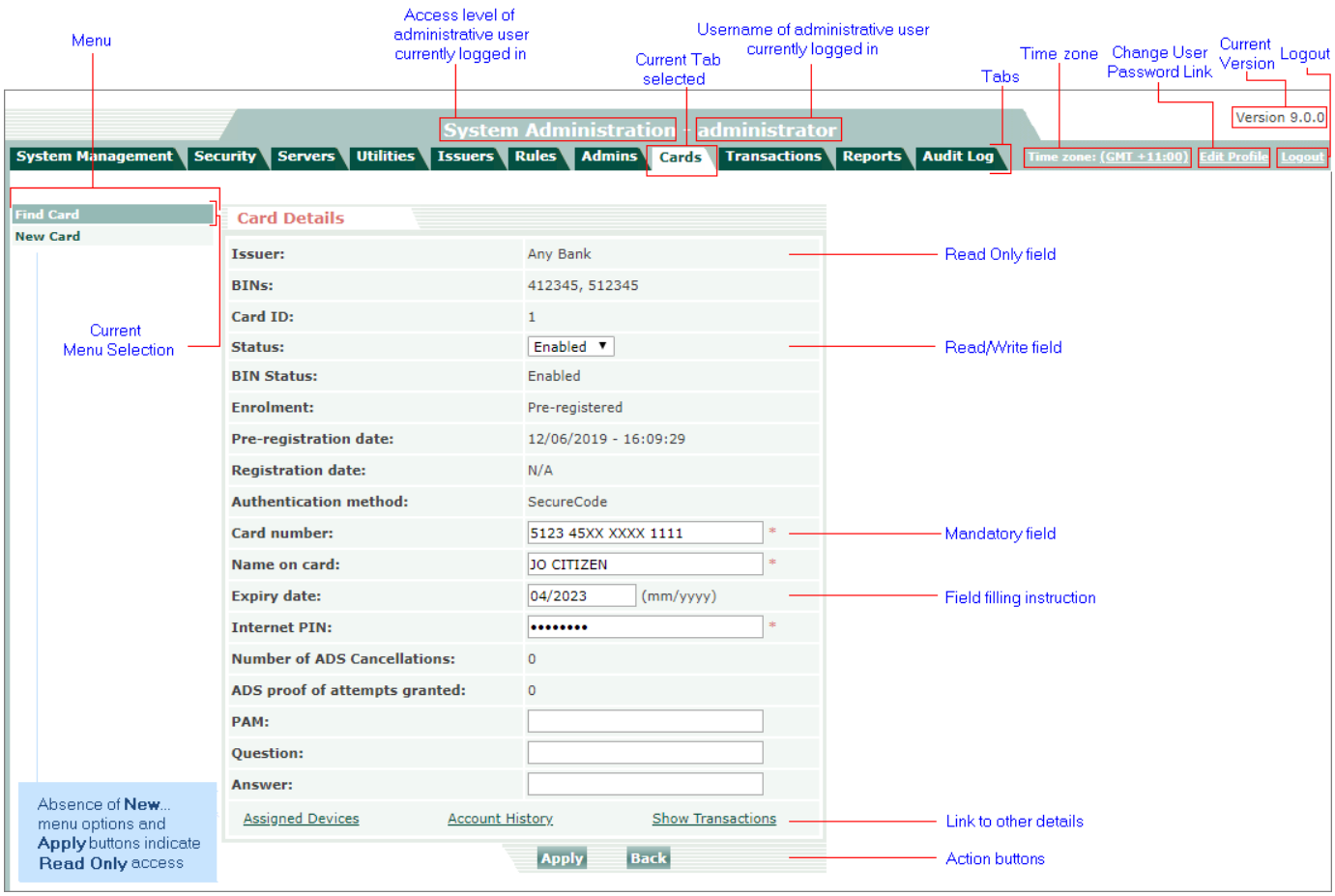
The appearance of the issuer administration pages is consistent throughout, with each being made up of a number of common components.

A banner area at the top of the screen displays the access level and the username of the user currently logged in; the system version; Time zone, Edit Profile and Logout links; and the main menu items as tabs.

Clicking on a menu tab displays the sub menu options on the left side of the page, with the first sub menu item highlighted.

Clicking on the required sub menu option displays the first page for that sub section.

The following example shows the key features of an typical ActiveAccess Administration page.



Issuer Administration Options

Use the menu tabs on the ActiveAccess Issuer Administration title bar to access the administration options. The complete set of options available is:

- **System Management** - set up and maintenance of system settings, issuer administration servers, issuers and issuer groups, issuer certificates, authentication management, issuer public and encryption keys, exchange configuration and archive management.
- **Security** - set up and maintenance of issuer signing certificates, Authentication History Server (AHS) certificates, CAAS (Remote Issuer) certificate, Issuer SDK certificate, Directory Server certificate, OOB Certificate, Risk Certificate, Decoupled Authenticator certificate, and trusted Certificate Authorities (CA).
- **Servers** - set up and maintenance of ACS, Administration, Authentication History Servers and CAAS Servers.
- **Utilities** - upload, manage and run system utilities.
- **Issuers** - set up and maintenance of specific member bank details including card details, rules, custom pages and key management.
- **Rules** - set up and manage business rules and the settings for risk based authentication.
- **Admins** - set up and maintenance of ActiveAccess administrative users.
- **Cards** - registration and maintenance of individual cards.
- **Transactions** - for accessing transactions, when required for cardholder support purposes, dispute resolution, etc.
- **Reports** - provides reports for card, enrolment and merchant activity, authentication, purchase volume, devices, admins and summary reports.
- **Audit Log** - provides a record of administrative user activity. It includes an extensive log of critical actions performed by the administrative staff.

Tip

The audit log section is available to system administrators and issuer administrators only. System administrators have access to an audit log of all events and issuer administrators have access to events relating only to their specific issuer or issuer group.

About System Management

System Administrators only

This section is used for setting up and maintaining system and ACS settings, Issuers and Issuer groups, settings for authentication devices, RBA, OOB and Decoupled Authenticator, Issuer public keys, exchange configuration and transaction record archiving. It has the following menu options:

- **Settings** - stores general settings for automatic logout idle time; maximum unsuccessful login attempts permitted; automatic unlock lag mktime; maximum number of concurrent logins allowed; and the password policy parameters for admin users.
- **ACS Settings** - Access Control Server and Remote Access Control Server related settings.
- **Issuer Management** - for setting up and maintaining Issuers and Issuer BIN ranges and viewing Issuer groups.
- **Group Management** - for setting up and maintaining Issuers groups and viewing group members
- **Authentication Management** - has settings for:
  - **Device Management** - for finding, setting up and managing devices used for authentication.
  - **Risk Management** - for managing risk chains and risk adapters used in 3DS2 risk based authentication.
  - **OOB Management** - for registering and managing the OOB adapters used for performing Out of Band (OOB) authentication challenges.
  - **Decoupled Authenticator Management** - for registering and managing the Decoupled Authenticator adapters used for performing Decoupled authentication challenges.
- **Public & Encryption Key Management** - for defining or updating Issuers' public and encryption keys, which are used to validate and decrypt registration API messages signed/ encrypted by the Issuer.

- **Exchange Configuration** - for displaying automatically downloaded external currency exchange rates and manually creating currency exchange values for rates not available on the automated list.
- **Archive Management** - for setting up automatic transaction record archiving.

Settings

System Management > Settings

This section is used to specify and maintain general configuration parameters such as automatic logout idle time; maximum unsuccessful logins permitted; automatic unlock lag time; maximum number of concurrent logins allowed; and the password expiry period, etc.

Use the following fields to complete this page:

- **Automatic Logout** time in minutes
  Acceptable range: 0 to 240
  Default: 20 min

  > **Warning**
  >
  > Setting this field to 0 disables the automatic logout mechanism and is not recommended.

  If an administrator account remains idle for the specified period of time it will be automatically logged out.
- **Maximum unsuccessful attempts** permitted for user logins.
  A greater number of unsuccessful login attempts will result in the administration account being locked, restricting further access to the account.
  Acceptable range: 0 to 9
  Default: 3

  > **Warning**
  >
  > Setting this field to 0 disables the automatic locking mechanism and is not recommended.

- **Automatic unlock** time in minutes.
  The amount of time after which a locked administrator account is automatically unlocked.
  Acceptable range: 0 to 1440
  Default: 0, which disables automatic unlocking such that all locked accounts have to be manually unlocked by another administrator user with the same or higher access level.
- **Maximum concurrent logins** permitted for MIA admin users
  Acceptable range: 0 to 9

> **Warning**
>
> Setting the number of concurrent logins to 0 will prevent more than one user logging in at the same time and is not recommended.

- The administration user **Password policy** is set using the following fields:
  - **Password expiry period** determines how often MIA administration users are required to change their MIA login password
    Acceptable range: 0 to 365
  - **Minimum password lifetime** determines the minimum number of days MIA administration users are required to wait before they can change their MIA login password again.
    Acceptable range: 0 to 90
  - **Minimum password length**
    Acceptable range: 0 to 32
    0 indicates no minimum length
  - **Minimum password numeric characters** required
    Acceptable range: 0 to 32
  - **Minimum password uppercase characters** required
    Acceptable range: 0 to 32
  - **Minimum password lowercase characters** required
    Acceptable range: 0 to 32
  - **Minimum password special characters** required
    Acceptable range: 0 to 32.

> **Note**
>
> The total number of characters entered for **Minimum password numeric characters**, **Minimum password uppercase characters**, **Minimum password lowercase characters** and **minimum password special characters** must be less than or equal to the **Minimum password length**.

- **Registration server URL** is the URL of the registration server used to send final and pre-registration requests to the registration server when issuers upload card data files.
- **Whitelisting server URL** is the URL of the whitelist server used to send add, remove, remove list, get merchant, and get merchant history requests to the whitelist server.

- **Time zone** is is displayed on the system administration menu bar, from where it can be modified at any time, as and when appropriate. The default time zone is set when the application is installed.
  All reports and results of searches will be based on the time zone specified on the menu bar at the time of the report or search.
- **Disable admin account if inactive for more than** a specified number of days.
  Acceptable range: 0 to 365.
  The system will disable an admin account if it has not been accessed for more than the specified number of days.
  To disable, set to 0.
- **AHS timeout** in seconds
  Acceptable range: 0 to 3600
  This defines the maximum amount of time the ACS will wait for the Authentication History Server to respond. If a response is not received within the expected time, the ACS will re-schedule the AHS transaction for a later time.
- **Show for the following access levels** checkboxes determine which administrator user roles are able to view **Card Number (plain text)** and **AAV/CAVV/AEVV**.
  By default, card numbers are masked and **AAV/CAVV/AEVV** is hidden.
- **Enable manual ACS restart** checkbox if you want to defer application of changes that require a restart to the next time the server is manually restarted.
  When this option is selected, if changes require the system to be restarted to take effect, you will be prompted that a restart is required. You can choose to defer the restart or select the option to Restart now.
- **Enable email notification messages** checkbox if you want the system to send email messages to administrators for two-factor authentication login or notifications such as expiring license keys.
  You will need to configure the mail server settings for this feature to work.
- **Mail server address**, **Mail server port**, **Mail server username**, **Mail server password** and **Mail server protocol** are used to record the address, username and password of an outgoing SMTP mail server.

> **Note**
>
> The sender of the notification messages will be the main administrator user (administrator). Make sure that you have specified a correct email address for this user (use **Edit Profile** link, while logged in as the administrator).

> **Note**
>
> You can test mail server settings by clicking on *Send Test Message* link. The link will appear once you have entered mail server settings and applied the changes.

- **Log level** determines the amount of information generated and routed to console and log file.
  Changes to log level take immediate effect.
  The options are:
  - **All:** includes any information that can be generated by the application.
  - **Debug:** information regarding more frequent and minor operations of the system or further.
  - **Info:** (Default) important information regarding the normal operation of the application or significant events.
  - **Warn:** warnings are minor errors that may not affect the operation of the system at all. For example a missing feature or component that may not affect the system if you are not planning to use its related functionality.
  - **Error:** log errors that may affect performance or operation of the system but do not necessarily prevent the system from operating. Logs incorrect behaviour of external components and systems outside the control of the application.
  - **Fatal:** logs severe problems, imminent system failure, application or component crash.
  - **Off:** Logging is disabled.

  ActiveAccess currently logs information in a subset of the above levels at **Fatal**, **Error**, **Warn**, **Info**, and **Debug** levels. Note that each higher level is inclusive of the messages of lower levels. For example when you set the log level to **Warn,** you will also see **Error** messages.

> **Database-related log level**
>
> Changing the value of the **Log level** field in **System Management > Settings** will apply it to all categories, except `DBSettings`, which is used for database-related logs. If required, you can update the `priority value` for all instances of the `DBSettings` category in **AA_HOME/log4j.xml**.
>
> The acceptable values for `DBSettings` log level are **OFF**, **INFO**, **DEBUG** and **TRACE**. The log level is set to **OFF** by default.

- ***Apply*** button to save changes.

ACS Settings

3D Secure 2 settings added

System Management > ACS Settings

The **ACS Settings** section is used to set local and remote (CAAS) Access Control Server options.

Use the following fields to set ACS settings:

- **ACS reference number**

  Displays a unique reference number provided by EMVCo to ActiveAccess.

- Select **Local** or **Remote (CAAS)** from the **Authentication server** drop down list.

3-D Secure 1 Settings

- **ACS URL** is the fully qualified URL of the Access Control Server's Payer Authentication (PA) processing page, as seen externally.

  The ACS URL specified here is passed to the merchant MPI as part of the ACS response to the Verify Enrolment (VEReq) message and is used by the merchant to transfer the session to the ACS for authentication of the cardholder.

  The default path for the ActiveAccess PA processing page is /acs/pa.

  > **Example**
  >
  > If you have installed ActiveAccess on the web server available on https://www.authenticationserver.com/ you should set the ACS URL to https://www.authenticationserver.com/acs/pa

- **Process timeout** in seconds
  Defines the maximum amount of time a cardholder has to complete their authentication. If the cardholder does not complete the authentication within the prescribed time, ACS returns a session timeout error.
  Acceptable range: 60 to 9000

- **Relative timeout** in seconds

Determines the amount of time a cardholder has to complete a single page, however, the total time to complete the whole authentication process may not exceed the **Process timeout**.

Acceptable range: 60 to 9000

3-D Secure 2 Settings

- **ACS challenge URL** is the fully qualified URL of the Access Control Server's Challenge (CReq) processing, as seen externally.

  The ACS URL specified here is passed to the 3DS Server as part of the ACS response to the Authentication Request (AReq) message and is used by the 3DS Requestor to transfer the session to the ACS for authentication of the cardholder.

  The default path for the ActiveAccess CReq processing page is /acs/ca.

  > **Example**
  >
  > If you have installed ActiveAccess on the web server available on https://www.authenticationserver.com/ you should set the ACS URL to https://www.authenticationserver.com/acs/ca

  The domain and protocol of the URL will be used for OOB device's WebSocket and callback URLs.

  > **Example**
  >
  > The WebSocket URL is: wss://www.authenticationserver.com/acs/oob-ws/
  >
  > The callback URL is: https://www.authenticationserver.com/acs/notify/

- **Initiate CReq timeout** in seconds

  Defines the maximum amount of time between the completion of the TLS handshake and the first CReq message sent to the ACS for processing. If the ACS does not receive any CReq within the prescribed time, it returns a transaction timeout error.

  Acceptable range: 15 to 60

- **Subsequent CReq timeout** in seconds

  Determines the amount of time a cardholder has to complete a single page in App mode. However, the total time to complete the whole authentication process may not exceed the **Process timeout**. If the cardholder does not complete a single page within the prescribed time, ACS returns a transaction timeout error.

Acceptable range: 300 to 1200

- **RRes timeout** in seconds

  Defines the maximum amount of time the Directory Server has to respond with RRes to the RReq sent by the ACS. If the Directory Server does not respond with RRes within the prescribed time, ACS returns a transaction timeout error.

  Acceptable range: 2 to 10

- **Browser authentication timeout** in seconds

  Determines the amount of time a cardholder has to complete a single page in Browser mode. However, the total time to complete the whole authentication process may not exceed the **Process timeout**. If the cardholder does not complete a single page within the prescribed time, ACS returns a transaction timeout error.

  Acceptable range: 300 to 1200

- **RReq retry interval** in seconds

  Failure to complete the initial connection and TLS handshake to the Directory Server for sending RReq results in an immediate retry. Upon second failure, the ACS will wait for the amount of time prescribed in RReq retry interval and retry to connect to the Directory Server.

  Acceptable range: 5 to 20

- **Process timeout** in seconds

  Defines the maximum amount of time a cardholder has to complete their authentication. If the cardholder does not complete the authentication within the prescribed time, ACS returns a transaction timeout error.

  Acceptable range: 315 to 1260

- ***Apply*** button to save changes.

Issuer Management

This section is used to define new issuers and issuer groups, or update existing information.

A group of issuers can be created for administration purposes. Issuer group determines which issuers users have access to; the administration group determines at which level they have access.

Links are provided to **ActiveDevice Settings** for assigning devices, and for creating a **New Issuer Group** or **New Issuer**.

A list of existing issuers and their group memberships is displayed. You can browse to view issuer and issuer group details by clicking on the ***Issuer Name*** and ***Group Membership*** links. The list can be filtered by Issuer Name, Issuer ID, BINs, Status and License Expiry period.

System Management > Issuer Management displays:

- A list of **Issuers**
- ***ActiveDevice Settings*** link to the **ActiveDevice Settings** page.
- ***New Issuer Group*** link to the **New Issuer Group** page.
- ***New Issuer*** link to the **New Issuer** page.

Use the following fields to limit the number of issuers displayed:

- **Issuer Name** (complete or partial) or leave empty to return all matching issuers
- **Issuer ID**, defaults to **All**
- **Issuer BINs** (comma separate multiple BINs)
- **Status** - **All** (default), **Enabled** or **Disabled**
- Select from the **License** key status drop down list:
  - All (default)
  - Valid
  - Expired
  - Expires in less than a month
  - Expires in 1 to 3 months

- Expires in 3 to 6 months
- Expires after due to expire in 1 to 6 months

- Click the **Go** button to display the new search results.

The following fields and links are displayed for each issuer:

- *Issuer Name* link to the **Issuer Details** page
- **Issuer ID**
- **BINs** - BIN numbers defined for the issuer
- *Group Membership* indicates the group to which the issuer belongs to. You can click on the group name to display the **Issuer Group Details** page.
- **Status** - Enabled or Disabled
- **License** - Shows the status of the issuer's license key

New Issuer Group

System Management > Issuer Management > New Issuer Group

This page is used to define a new issuer group and to assign issuers to that group.

Use the following fields to add a new issuer group:

- **Name** of the issuer group. It is a good idea to use the word "group" as part of the name for example "ABC Group".

  The system will automatically assign a **Group ID** to the issuer group.
- Optionally specify a parent by selecting from the **Parent group** drop down list.

  This allows you to build a hierarchy of issuers and groups to suit your administration requirements.
- **ACS URL** - the system allows a separate URL to be created for each issuer group. If a separate URL is required, it should be entered here.
- **ACS Challenge URL** - the system allows a separate URL to be created for each issuer group. If a separate URL is required, it should be entered here.

> **Note**
>
> Any changes to this URL will require changes to OOB device's WebSocket and callback URLs.

- **Uses confirmation** - Indicates if the Issuer uses the confirmation method. Defaults to **No**.

  In the Enrolment component, if **Uses confirmation** is enabled, the cardholder will be taken through the sign up process. If set to disabled, the registration status of the card will be checked and displayed to the cardholder.

  When **Activation During Shopping** is enabled, if the cardholder is **pre-registered** and **Uses confirmation** is **No**, the cardholder is required to create a 3-D Secure password (VbV password / Mastercard SecureCode / JSecure password / American Express SafeKey / ProtectBuy password) to use in the authentication process.

  If **Uses confirmation** is **Yes**, the cardholder's existing registration data is used in the authentication process, instead of requiring a new 3-D Secure password (VbV password / Mastercard SecureCode / JSecure password / American Express SafeKey / ProtectBuy password) to be created.

  Select one or more issuers or groups to add to the group from the **Issuer Members** list or the **Group Members** list. Use the **Add >>** button to add the issuers or child groups to the **Selected** list.

  You can use the **<<Remove** button to remove issuers or child groups from the issuer group.

- **SecureCode MAC algorithm** used in conjunction with **SecureCode** transactions (3DS1 only).

> **Note**
>
> This will be used only if the issuer's **Use parent keys** option is enabled.

- **IAV generation algorithm** used in conjunction with Mastercard Identity Check transactions (3DS2 only).

> **Note**
>
> This will be used only if the issuer's **Use parent keys** option is enabled.

- **Visa CEMEA region** used in conjunction with **Verified by Visa** and **Visa Secure** transactions.

> **Note**
>
> - This will be used only if the issuer's **Use parent keys** option is enabled.
> - When **Visa CEMEA region** is set to **Yes**, then **CAVV format** will be updated to **U3V7**.

- **Verified by Visa CAVV format** used in conjunction with **Verified by Visa** transactions (3DS1 only).

> **Note**
>
> This will be used only if the issuer's **Use parent keys** option is enabled.

- **Visa Secure CAVV format** used in conjunction with **Visa Secure** transactions (3DS2 only).

> **Note**
>
> This will be used only if the issuer's **Use parent keys** option is enabled.

- **Group Members** - Use the **Add >>** and **<< Remove** buttons to add or remove child groups that should belong to the group.
- **Issuer Members** - Use the **Add >>** and **<< Remove** buttons to add or remove issuers that should belong to the group.
- **Use parent certificate, public and encryption keys** option indicates that the group does not have a certificate of its own and will use the parent group's certificate and registration API public key and encryption key. This option is only enabled if you have specified a parent group. Enabling the parent certificate will automatically enable the use parent keys options.
- **Use parent keys** option to indicate that the group does not have any keys of its own and will use the parent group's keys. This option is only enabled if you have specified a parent group.
- ***Apply*** button to save changes.

> **Note**
>
> Once the issuer group has been created, you may optionally specify a separate **ACS URL** for it by editing the Issuer Group Details.

Issuer Group Details

This page is used to view/edit issuer group details and assign issuers to, or remove issuers from, the issuer group.

System Management > Issuer Management > Issuer Group Details - fields

Info

See the **New Issuer Group** section of this document for additional information on these fields.

- **Group ID** is a unique identifier, which is used by the system in order to reference the group. Group ID cannot be changed.
- **Name** of the issuer group
- **Parent group** - you can optionally define a parent group in order to create a hierarchy of groups and issuers to suit your administration requirements.
- **ACS URL** - the system allows a separate URL to be created for each issuer group. If a separate URL is required, it should be entered here.
- **ACS Challenge URL** - the system allows a separate URL to be created for each issuer group. If a separate URL is required, it should be entered here.

Note

Any changes to this URL will require changes to OOB device's WebSocket and callback URLs.

- **Uses confirmation** - Indicates if the Issuer uses the confirmation method.

  The confirmation method is a process allowing cardholders with an enrolment status of "Pre-registered" to utilise their pre-registration account information, instead of creating a new 3-D Secure password, to perform 3-D Secure authentication.
- **SecureCode MAC algorithm** to be used in conjunction with **SecureCode** transactions (3DS1 only).

Note

This will be used only if the issuer's **Use parent keys** option is enabled.

- **IAV generation algorithm** used in conjunction with **Mastercard Identity Check** transactions (3DS2 only).

Note

This will be used only if the issuer's **Use parent keys** option is enabled.

- **Visa CEMEA region** used in conjunction with **Verified by Visa** and **Visa Secure** transactions.

> **Note**
>
> - This will be used only if the issuer's **Use parent keys** option is enabled.
> - When **Visa CEMEA region** is set to **Yes**, then **CAVV format** will be updated to **U3V7**.

- **Verified by Visa CAVV format** used in conjunction with **Verified by Visa** transactions (3DS1 only).

> **Note**
>
> This will be used only if the issuer's **Use parent keys** option is enabled.

- **Visa Secure CAVV format** used in conjunction with **Visa Secure** transactions (3DS2 only).

> **Note**
>
> This will be used only if the issuer's **Use parent keys** option is enabled.

- **Group members** - Child groups that belong to the group are listed in the **Selected** list. Other groups (not belonging to any other group) are listed in the **Available** list. Use the **Add >>** and **<< Remove** buttons to change the child groups that belong to the group.
- **Issuer members** - issuers that belong to the group are listed in the **Selected** list. Other issuers are listed in the **Available** list. Use the **Add >>** and **<< Remove** buttons to change the issuers that belong to the group.
- **Use parent certificate, public and encryption keys** - Selecting this option indicates that the issuer group does not have a certificate of its own and will use the parent group's certificate, registration API public key and encryption key. The option is only enabled if you have specified a parent group. Enabling the parent certificate will automatically enable the use parent keys options.

> **Note**
>
> Enabling this option will remove the issuer group certificate (if it has one) from the system. You cannot retrieve the certificate once removed.

> **Note**
>
> When you disable this option, the issuer group will no longer use the parent's certificate. You need to create a certificate request for the issuer group and have it signed by the appropriate CAs.

> **Warning**
>
> It is recommended that you make a decision to enable or leave this option disabled at the time of creating the issuer group to avoid the administration overhead of changing this option later.

- **Use parent keys** - Selecting this option indicates that the issuer group does not have any keys of its own and will use the parent group's keys. The option is only enabled if you have specified a parent group.

  Selecting this option indicates that the issuer group does not have any keys of its own and will use the parent group's keys. The option is only enabled if you have specified a parent group.

> **Note**
>
> Changing this option invalidates the issuer group existing certificate. You either need to enable the 'Use parent certificate' option or create a new certificate request, and have it signed by the appropriate CAs.

Note

Enabling this option will delete the issuer group keys from the local HSM. Deleting keys is irreversible unless you have previously backed them up. The following keys will be removed from the local HSM, where < group_id > is the issuer group's unique identifier as shown in the issuer group details:

- SPA< group_id >
- VbVA< group_id >
- VbVB< group_id >
- JCBA< group_id >
- JCBB< group_id >
- MSCA< group_id >
- MSCB< group_id >
- SKA< group_id >
- SKB< group_id >
- DCA< group_id >
- DCB< group_id >
- RSAVbV< group_id >_pub
- RSAVbV< group_id >_pri
- RSAMSC< group_id >_pub
- RSAMSC< group_id >_pri
- RSAJCB< group_id >_pub
- RSAJCB< group_id >_pri
- RSASK< group_id >_pub
- RSASK< group_id >_pri
- RSADC< group_id >_pub
- RSADC< group_id >_pri
- RSADEVICE< group_id >_pub
- RSADEVICE< group_id >_pri

If you are using other HSMs in your system, you also need to remove these keys from those HSMs to keep them synchronised. You also need to update any other party who may use these keys for verification of AAV (UCAF) or CVV (CAVV).

Warning

Disabling this option will create new keys for the issuer group, where < group_id > is the issuer group's unique identifier as shown in the issuer group details. The following keys will be created on the local HSM:

- SPA< group_id >
- VbVA< group_id >
- VbVB< group_id >
- JCBA< group_id >
- JCBB< group_id >
- MSCA< group_id >
- MSCB< group_id >
- SKA< group_id >
- SKB< group_id >
- DCA< group_id >
- DCB< group_id >
- RSAVbV< group_id >_pub
- RSAVbV< group_id >_pri
- RSAMSC< group_id >_pub
- RSAMSC< group_id >_pri
- RSAJCB< group_id >_pub
- RSAJCB< group_id >_pri
- RSASK< group_id >_pub
- RSASK< group_id >_pri
- RSADC< group_id >_pub
- RSADC< group_id >_pri
- RSADEVICE< group_id >_pub
- RSADEVICE< group_id >_pri

If you are using other HSMs in your system, you also need to export these keys to those HSMs to keep them all synchronised. You also need to update any other party who may use these keys for verification of AAV (UCAF) or CVV (CAVV).

Tip

It is recommended that you make a decision to either **enable** or leave this option **disabled** at the time of creating the issuer, to avoid the administration overhead of changing this option later.

> **Info**
>
> Refer to New Issuer Group for additional information on these fields.

- ***Apply*** button to save changes.

> **Note**
>
> A group cannot be removed if it has other groups or Issuers belonging to it.

New Issuer

Use this page to define a new issuer and optionally assign the issuer to an issuer group.

System Management > Issuer Management > New Issuer - fields

- **Status** of **Not Registered** is automatically assigned to new issuers by the system and cannot be changed until you have obtained a license key from GPayments.
- Enter the **Name** of the Issuing bank or financial institution.

  You must enter a name that is unique in the issuer system.
- Enter an optional **Password** for the Issuing bank or financial institution.

  This password is used for authentication of issuer connection to ActiveAccess via UAC. This is in addition to the verification of issuer's client authentication and may be left empty if the extra verification is deemed to be unnecessary.
- **ACS URL** - the system allows for a separate URL to be created for each Issuer. If a separate URL is required, it should be entered here.
- **Show extended account information** - Select **Yes** to display all cardholder pre-registration account information, on the card details page, created during the cardholder's Pre-registration with the system. When this option is disabled, only basic cardholder information is displayed on this page.
- **Uses confirmation** - Select **Yes** or **No** to indicate if the Issuer uses the confirmation method.

  The confirmation method is a process allowing cardholders with an enrolment status of "Pre-registered" to utilise their pre-registration account information, instead of creating a new 3-D Secure password, to perform 3-D Secure authentications.

- **Event Logging** - **Disabled** by default. Select **Enabled** to indicate event logging is required or **Enable V+ compatible** to indicate event logging is required, and the maximum number of Activation During Shopping opt-out events reported to the issuer is 9.

  This feature allows issuers to download cardholder events through Registration Server Notification messaging. A Notification is a record of a single cardholder event. Each event is stored in ActiveAccess and a record is logged in the event a cardholder completes their registration, opts-out of Activation During Shopping or locks their account.

- If the issuer is to be assigned to an issuer group, select the group from the **Parent group** drop down list.
- If you have specified a parent group:
  - You may select the **Use parent certificate, public and encryption keys** option to indicate that the issuer does not have a certificate of its own and will use the parent group's certificate and registration API public key and encryption key.
  - You may select the **Use parent keys** option to indicate that the issuer does not have any keys of its own and will use the parent group's keys.
- ***Apply*** button to save changes.

Tip

Once you have created the Issuer record a confirmation message will be displayed:

Please note down the **Issuer ID** and **Issuer Name**, and send them to GPayments in order to request a license key for the newly generated issuer.

Note

Once the new issuer has been created, you may optionally specify a separate **ACS URL** for it by editing the Issuer Details.

Info

Refer to Issuer Details for additional information on these fields.

Issuer Details

This page is used to view/ edit issuer details and assign the issuer to, or remove the issuer from an issuer group.

System Management > Issuer Management > Issuer Details (Local and Remote Issuers)

Use the following fields to view / edit issuer details:

> **Note**
>
> Not all fields will be visible to all issuers, depending on issuer or issuer group settings.

- **Issuer ID** is a unique identifier, which is used by the system in order to reference the issuer. Issuer ID cannot be changed. Issuer ID is used in a number of situations such as requesting license key for the issuer, sending pre-registration and final registration messages and also forms part of the unique URL which is used for the issuer enrolment site.
- **Status** - **Enabled** or **Disabled**, if the issuer is registered. Prior to the issuer obtaining a valid license key the Status is displayed as **Not Registered** and cannot be changed.

  > **Note**
  >
  > An issuer account that does not have a valid licence key is practically disabled. This makes most functions unavailable to the issuer including the enrolment, registration and authentication of cardholders.

- **Name** of the issuing bank or financial institution.

  > **Note**
  >
  > This field forms part of the issuer licence key information. You will need to re-apply for a licence key if you change this field.

- **Password** for the Issuing bank or financial institution.

  This password is used for authentication of issuer connection to ActiveAccess via UAC. This is in addition to the verification of issuer's client authentication and may be left empty if the extra verification is deemed to be unnecessary.

- **ACS URL** - the system allows a separate URL to be created for each issuer. If a separate URL is required, it should be entered here.

- **ACS Challenge URL** - the system allows a separate URL to be created for each issuer. If a separate URL is required, it should be entered here.

> **Note**
>
> Any changes to this URL will require changes to OOB device's WebSocket and callback URLs.

- **Show extended account information -** Select **Yes** to display all cardholder data as sent by the Registration API messages in the card details page. When this option is disabled, only basic cardholder information is displayed.
- **Allow issuer to access rules -** Select **Yes** or **No** to indicate if the Issuer can access the business rules functionality.

  Business rules are configurable settings which provide issuers control over the customer process during the 3-D Secure transactions. Rules can be configured using a 3-D Secure transaction's parameters such as the Transaction Amount, the Merchant ID, Merchant Name, Acquirer BIN or Merchant Country.

> **Note**
>
> This feature is only available, if the custom pages are rule-compatible.

- If **Allow issuer to access rules** is set to **Yes**, then **Grant Access to Business Admin** and **Grant Access to Helpdesk** checkboxes allow you to grant Business Admin and / or Helpdesk users access to the Rules section. Whether these users have *read only* or *full* access is determined by their Admins settings
- **Authentication Server - Local** or **Remote (CAAS)**
- If **Remote (CAAS)** is selected, **CAAS server** will be displayed, to allow selection of the already configured remote authentication servers.
- If **Authentication server** is set to **Remote (CAAS)**, optionally select the **Risk engine integration** checkbox if the authentication process is to be integrated with the issuer's risk engine.
- **Risk chain** - Select an already configured Risk Chain from the drop down list to enable Risk-Based authentication for the issuer.
- **ACS interface** - Select **Native** (default) or **HTML** from the drop down list.

Identifies the ACS interface for presenting the challenge to the cardholder: Native UI or HTML UI. In SDK mode, if the supported interface is not specified in the AReq, the ACS uses the interface that is selected in this field.

- **Uses confirmation** - Indicates if the Issuer uses the confirmation method. Defaults to **No**.

  The confirmation method is a process allowing cardholders with an enrolment status of "Pre-registered" to utilise their pre-registration account information, instead of creating a new 3-D Secure password, to perform 3-D Secure authentications.

- **Visa CEMEA region** - Visa CEMEA require that a CAVV be generated and returned in all PARes, ➕ ARes and ➕ RReq messages regardless of the authentication status. Set this field to **Yes**, if this functionality is required.

> **Note**
>
> - When **Use parent keys** option is enabled, the parent's Visa CEMEA region will be used.
> - When **Visa CEMEA region** is set to **Yes**, then **CAVV format** will be updated to **U3V7**.

- **SecureCode MAC algorithm** - determines the algorithm which is used for calculation of AAV field for SecureCode transactions. By default HMAC algorithm is used. You may change this to CVC2 if required (3DS1 only).

> **Note**
>
> - The application generates two 3DES keys, when a CVC2 option is selected for the first time: MSCA< issuer_id > and MSCB< issuer_id >.
> - ➕ When **Use parent keys** option is enabled, the parent's SecureCode MAC algorithm will be used.

- **IAV generation algorithm** - determines the algorithm which is used for calculation of AAV field for Mastercard IDC transactions. By default, **DS Transaction ID + PAN** algorithm is used. You may change this to `PAN`, `DS Transaction ID`, `Coded Amount + DS Transaction ID + PAN`, or `Merchant Name + Coded Amount + DS Transaction ID + PAN` if required (3DS2 only).

> **Note**
>
> When **Use parent keys** option is enabled, the parent's SecureCode MAC algorithm will be used.

- **Verified by Visa CAVV format** used in conjunction with **Verified by Visa** transactions (3DS1 only).

> **Note**
>
> When **Use parent keys** option is enabled, the parent's Verified by Visa CAVV format will be used.

- **Visa Secure CAVV format** used in conjunction with **Visa Secure** transactions (3DS2 only).

> **Note**
>
> When **Use parent keys** option is enabled, the parent's Verified by Visa CAVV format will be used.

- **Force cardholders to use device** - if Device Authentication is available, select **Yes** to force cardholders to register their authentication device during the 3-D Secure authentication process. Select **No**, to provide cardholders with a link to allow them to register their authentication device.
- **Event Logging** - **Disabled** (default) or **Enabled** to indicate event logging is required; or **Enable V+ compatible** to indicate event logging is required and the maximum number of Activation During Shopping opt-out events reported to the issuer is 9.
- **Parent group** - select the group to which the Issuer belongs, if any.

> **Note**
>
> The issuer can only be assigned to a single group; however the group itself can belong to another group. This enables you to create a hierarchy of issuers and groups to suit your administration needs.

- **License key** - copy license key provided by GPayments and click ***Apply*** button. The **Status** will then change to **Enabled**.
  - **License status** - once you have entered a valid license key the license status will display the validity period for the key (e.g. **License key is valid until 01/03/2019)**, the 3-D Secure authentication protocol version, and whether Risk, OOB, Decoupled Authenticator, NPA, APP, and 3RI features are supported.

> **Note**
>
> If the licence key is not present, invalid or expired, the issuer account is practically disabled. This makes most functions unavailable to the issuer including authentication of cardholders, registration and whitelisting.

- **Issuer BINs** - Use the **BIN Management** link to add, edit, delete, enable and disable one or more BINs for the issuer and specify if device authentication or  whitelisting is available for cards that belong to the specified BIN.
- **Use parent certificate, public and encryption keys** - Selecting this option indicates that the issuer does not have a certificate of its own and will use the parent group's certificate, registration API public key and encryption key. The option is only enabled if you have specified a parent group. Using the parent certificate is only possible if you have also chosen to use the parent keys. Enabling this option automatically enables the use parent keys.

> **Note**
>
> - Enabling this option will remove the issuer's certificate (if it has one) from the system. You cannot retrieve the certificate once removed.
> -  Enabling this option will disable the issuer's CAVV/IAV related configuration and use parent's.
> - When you disable this option, the issuer will no longer use the parent's certificate. You need to create a certificate request for the issuer and have it signed by the appropriate CAs.

> **Tip**
>
> It is recommended that you make a decision to enable or leave this option disabled at the time of creating the issuer to avoid the administration overhead of changing this option later.

- **Use parent keys** - Selecting this option indicates that the issuer does not have any keys of its own and will use the parent group's keys. The option is only enabled if you have specified a parent group.

> **Note**
>
> - Changing this option invalidates the issuer's existing certificate. You either need to enable the 'Use parent certificate' option or create a new certificate request, and have it signed by the appropriate CAs.
> -  Enabling this option will disable the issuer's CAVV/IAV related configuration and use parent's.

Warning

Enabling this option will delete the issuer's keys. Deleting keys is irreversible unless you have previously backed them up. The following keys will be removed, where < issuer_id > is the issuer's unique identifier as shown in the issuer details:

- SPA< issuer_id >
- VbVA< issuer_id >
- VbVB< issuer_id >
- JCBA< issuer_id >
- JCBB< issuer_id >
- MSCA< issuer_id >
- MSCB< issuer_id >
- SKA< issuer_id >
- SKB< issuer_id >
- DCA< issuer_id >
- DCB< issuer_id >
- RSAVbV< issuer_id >_pub
- RSAVbV< issuer_id >_pri
- RSAMSC< issuer_id >_pub
- RSAMSC< issuer_id >_pri
- RSAJCB< issuer_id >_pub
- RSAJCB< issuer_id >_pri
- RSASK< issuer_id >_pub
- RSASK< issuer_id >_pri
- RSADC< issuer_id >_pub
- RSADC< issuer_id >_pri

You also need to update any other party who may use these keys for verification of AAV (UCAF) or CVV (CAVV).

Note

Disabling this option will create new keys for the issuer. The following keys, where < issuer_id > is the issuer's unique identifier as shown in the issuer details, will be created:

- SPA< issuer_id >
- VbVA< issuer_id >
- VbVB< issuer_id >
- JCBA< issuer_id >
- JCBB< issuer_id >
- MSCA< issuer_id >
- MSCB< issuer_id >
- SKA< issuer_id >
- SKB< issuer_id >
- DCA< issuer_id >
- DCB< issuer_id >
- RSAVbV< issuer_id >_pub
- RSAVbV< issuer_id >_pri
- RSAMSC< issuer_id >_pub
- RSAMSC< issuer_id >_pri
- RSAJCB< issuer_id >_pub
- RSAJCB< issuer_id >_pri
- RSASK< issuer_id >_pub
- RSASK< issuer_id >_pri
- RSADC< issuer_id >_pub
- RSADC< issuer_id >_pri

You also need to update any other party who may use these keys for verification of AAV (UCAF) or CVV (CAVV).

Tip

It is recommended that you make a decision to enable or leave this option disabled at the time of creating the issuer to avoid the administration overhead of changing this option later.

- **Email Address** - may be used in OTP emails (parameter: $IssuerEmail - max 128 char).
- **Customer service phone number** may be used in OTP emails (parameter: $ServicePhoneNumber - max 32 char).

- **ActiveDevice Settings** - used to assign one or more device types to a selected issuer and specify device sharing rules.
- ***Apply*** button to save changes.

BIN Management

This section is used to manage BINs of a specified issuer. Each BIN provides a link to allow you to edit the BIN, the status of Device over 3-D Secure, the status of whitelisting or the status of the BIN. BINs can be selected and deleted from the system using the **Delete** button. Only BINs which have no cards assigned to them on the system can be deleted. The **Enable** and **Disable** buttons can be used to change the status of the BIN. New BINs can be added for the issuer through the *Add BIN* link, and device authentication and/or whitelisting can be made available for cards that belong to the specified BIN.

System Management > Issuer Management > Issuer Details > BIN Management > Add BIN

Use the following fields to add a BIN:

- **Issuer** is displayed and cannot be changed
- **BIN**
- **Device over 3-D Secure** - **Disabled** or **Enabled** to specify if device authentication is available for cards that belong to this BIN
- **Whitelisting** - **Disabled** or **Enabled** to specify if the process of placing 3DS Requestors on the cardholders' trusted beneficiaries list is available for cards that belong to this BIN.
- **Status** - **Disabled** or **Enabled** to specify the availability of the 3-D Secure service for cards that belong to this BIN. Cards with a **Disabled** BIN cannot be enrolled, registered or authenticated
- ***Apply*** button to save changes.

ActiveDevice Settings

This section is used to assign one or more device types to a selected issuer and specify device sharing rules. An issuer may choose to share devices with none, all, or a selected number of issuers and issuer groups.

> **Note**
>
> Device parameters for SMS and email devices are issuer specific and these devices are not shared between issuers and issuer groups. However, the same mobile numbers / email addresses can be registered for different issuers. ActiveAccess treats SMS / email devices that have the same mobile numbers / email addresses as independent devices.

System Management > Issuer Management > Issuer Details > ActiveDevice Settings

Use the following fields to view / edit ActiveDevice settings:

- **Issuer**
- **Supported devices** - authentication devices accepted by the Issuer are listed in the **Selected** list. Other available devices not currently selected by the issuer are listed in the **Available** list. Use the **Add** and **Remove** buttons to change the tokens assigned to the issuer.

  > **Warning**
  >
  > If you remove any of the supported devices, the cardholder will no longer be able to use that device and transactions may fail.

- **Allow sharing device with** allows the issuer to share its devices will all, none or a selected list of issuers and groups.
- ***Apply*** button to save changes.

To view device parameter details, click the **Device Parameters** button.

The **Edit Device Parameters** page will be displayed.

> **Warning**
>
> For hardware and software token devices, changing device parameters may adversely affect the authentication of users. Such device parameters must be left as default unless absolutely necessary. You must consult with the device manufacturer before making any changes to these parameters.

> **Note**
>
> For information on default device parameters, go to Device Management.

Edit Device Parameters

The first available Device type for the selected Issuer is displayed.

Use the following fields to edit Device Parameters:

- **Device type** This parameter can be left as the default or customised for the selected issuer.

  The available device types are:

  - Backup Device
  - Decoupled Authenticator
  - Email
  - OOB
  - SMS
  - VASCO
- **Use device's default parameters** - if this option is selected, it indicates that the issuer will use the Default Device Parameters for the selected device.

  Deselect the checkbox to customise the device parameters. If the checkbox is already deselected, you can reset the parameters to the default by selecting it.

> **Note**
>
> For full details of device parameters, refer to Default Device Parameters.

The following fields are additional to the configurable fields in Default Device Parameters:

- **Device type**: SMS
  - **Available SMS Centres** - use the **Add >>** and **<< Remove** buttons to select the appropriate SMS Centres.
- **Device type**: OOB
  - **Available OOB adapters** - use the **Add >>** and **<< Remove** buttons to select the appropriate OOB Adapters
- **Device type**: Decoupled Authenticator
  - **Available Decoupled Authenticator adapters** - use the **Add >>** and **<< Remove** buttons to select the appropriate Decoupled Authenticator Adapters.

Group Management

System Management > Group Management

This section is used to set up and maintain issuer groups. It provides access to the same functions as the Issuer Management section, but from an issuer group perspective.

Organising related issuers in a group can greatly reduce the issuer administration overhead. Groups can have their own keys (AAV key [1], CVC2 keys [2], CVV keys [3], IAV HMAC keys [4] and signing key [5]) and certificates. An issuer can be configured to use the parent group's keys to reduce the number of keys generated and as a result, also reduce the overhead of key management tasks for synchronizing multiple hardware security modules. An issuer may also be configured to use the parent group's certificate in order to reduce the overhead of certificate management and renewal.

A list of issuer groups and their issuer and group members is displayed. You can browse to view issuer group and issuer details by clicking on the **Group Name**, **Issuer Members** and **Group Members** links.

The following fields and links are displayed:

- **Group Name** links to **Group Details** page
- **Group ID**
- **Group Members** links to **Issuer Group Details** page, shows the issuer groups and issuers that belong to this group
- **Issuer Members** links to **Issuer Details** page, shows the license key and certificate details for the selected issuer
- **New Issuer Group**
- **New Issuer**

---

1. 192-bit generic key, used in AAV HMAC calculation for SecureCode transactions
2. Pair of DES or 3DES keys, used in AAV CVC2 calculation for SecureCode transactions
3. Pair of DES or 3DES keys, used in CVV calculation for VbV and Visa Secure transactions
4. 256-bit generic key, used in IAV HMAC calculation for IDC transactions
5. RSA key pair, used for signing the PARes messages and ACSSignedContent in ARes for app-based transactions.

About Authentication Management

The **Authentication Management** section is used for:

- **Device Management** - for finding, setting up and managing devices used in the authentication process.
- **Risk Management** - for managing risk chains and risk adapters used in 3DS2 risk based authentication.
- **OOB Management** - for registering and managing the OOB adapters used for performing Out of Band (OOB) authentication challenges.
- **Decoupled Authenticator Management** - for registering and managing the Decoupled Authenticator adapters used for performing Decoupled authentication challenges.

Device Management

*The **Device Management** section is grouped with **OOB Management, Decoupled Authenticator Management** and **Risk Management** in the **Authentication Management** section.*

This section is used for finding devices, updating device status, uploading hardware token device initialization seed files, and configuring default device parameters.

Note

The term 'devices' is used as a generic term for both devices used for authentication and authentication methods. It includes:

- Hardware and software tokens
- Authentication methods such as OTP with SMS or email
- A standalone backup token

Info

Device files for hardware tokens are provided by the device manufacturer and contain information that uniquely identifies each authentication device and can be used to verify the tokens / passwords generated by that device. Each hardware token device is identified by a serial number. The serial number is determined by the device manufacturer and must be unique per device type.

Once a seed file is uploaded into the system, cards can be assigned to devices by linking device serial numbers with card accounts. Once an account is linked with a device serial number, the card enrolment process is complete.

System Management > Authentication Management > Device Management displays

- A list of recently **uploaded device seed files** for hardware tokens. By default the system displays the seed files uploaded in the last 10 days.
- **Edit Default Device Parameters**
- **Upload File** to schedule a new job
- **Find Device** to view or edit the details of each device.

Use the following fields to limit the upload files displayed:

- **Issuer**

- **Device Type**
- **From** and **To Date**
- **Refresh** button to display the new list.

The following fields and links are displayed:

- **Job number** link to the **Job Details** page to view job details including any error message or warnings.
- **Issuer name** (owner of the devices)
- **File Name**
- **Device type**
- When the upload was **Started** and **Finished**
- Number of **Attempts** before the upload was finished
- **Status** of the job: get the current status by pressing the refresh button

Job Details

This page displays details of the seed file upload, including any error messages or warnings, for the job selected on the **Upload File** page.

System Management > Authentication Management > Device Management > Job Details displays

- **Issuer** name
- **Job number**
- **Uploaded** - date and time when the file was first uploaded
- **Device type**
- **File Name**
- **Start** and **Finish** date and time the job
- **Attempts** before the upload was finished
- **Status**
- **Error message**, if any.
- **Error details**
- **Warnings**

Edit Default Device Parameters

System Management > Authentication Management > Device Management > Edit Default Device Parameters

Each device has its own set of device parameters. In the case of hardware tokens, these are manufacturer-defined parameters, such as VASCO, supported by adding additional libraries and installing vendor specific drivers. Other devices, such as SMS and Email are virtual devices natively supported by ActiveAccess.

Device parameters can be customised per issuer. By default this customisation is disabled, such that all issuers use the default device parameters.

Use the following fields to edit default device parameters:

- **Device type**

  The options are:

  - Backup Device
  - Email
  - OOB (Out of Band)
  - SMS
  - VASCO

SMS

System Management > Authentication Management > Device Management > Edit Default Device Parameters - SMS

SMS is a virtual device natively supported by ActiveAccess. This is in contrast to some third party devices such as VASCO which are supported by adding additional libraries and installation of vendor specific drivers.

The SMS device can be used as a backup device.

The SMS device parameters page is where the administrator can setup the system for sending SMS messages. ActiveAccess supports SMPP-API-0.3.9.1 (Short Message Peer to Peer) protocol for sending SMS messages to an SMS gateway, also known as an SMSC (Short

Message Service Centre). The SMS gateway is normally provided by the business section of your preferred telecommunications company.

The connection to the SMSC must be over TCP/IP. The details of connection to the SMSC will be provided by your telecommunications company.

Use the following fields to edit SMS Device Parameters:

- **Device type** - **SMS**
- **SMS token type** - ActiveAccess can generate two types of SMS tokens:
  - **Instant** - the system generates one SMS token per authentication. The token is generated and sent to the cardholder's mobile phone, after the verify enrolment request is received by ActiveAccess.
  - **Batch** - the cardholder receives a batch of SMS tokens beforehand. The batch SMS message contains a batch reference number and a list of generated tokens, each identified by a letter of the alphabet. The cardholder is then asked to enter a token that corresponds with a specific letter of the alphabet as shown on the authentication page. With batch SMS, up to 15 tokens can be sent in a single SMS message and hence reduce the cost of sending SMS tokens. The system generates another set of tokens and sends them to the cardholder when the last token for the current batch is used.
- **Batch SMS lifetime** - determines the validity period of batch SMS tokens in days (acceptable range is 0 to 365). Batch tokens will be valid for the period specified by this option. The default is 30 days.
- **Instant SMS lifetime** - determines the validity period of instant SMS tokens in minutes (acceptable range is 0 to 60). Following Instant tokens will be valid for the period specified by this option. The default is 15 minutes. You should consider the mobile network delay for sending SMS messages and provide sufficient time for the cardholder to enter the token.
- **SMS token length** - determines the number of digits in the token generated (acceptable range is 6 to 10). The default is 6 digits.
- **Number of tokens in each batch** - determines the number of tokens included in a batch. The default is 10.

  An SMS message on a GSM network may contain up to 160 characters, while the limit for a CDMA network is between 120 to 153 characters. The system limits the maximum number of tokens based on the CDMA's lower limit of 120 characters.
- **Maximum unsuccessful attempts to send an SMS** - (acceptable range is 0 to 9) if sending an SMS message fails due to network or application errors, such as connection problems to

the SMSC or receiving an invalid response from the SMS, the system attempts to resend the SMS message up to the number of times specified by this option. The default value is 5. If all attempts for delivering fail, an error is reported back to the administration user.

- **Maximum number of SMSs sent per authentication session** - (acceptable range is 0 to 99) determines the number of times that a new SMS OTP can be requested by the cardholder during each authentication session. The default value is 3. If the limit is reached, the authentication fails.
- **Accept mobile numbers of** - Select the country name that you would like to accept as SMS mobile number. Select 'All' if you would like to accept all international mobile numbers.
- **Restrict mobile number** - Turn this option on if you want to specify a mobile number format. Enter the required format, eg. ##########, 61#########, 0061#########. Allowed characters are 0-9, '#', '(', ')', '-' and space. Please note that the mobile number, excluding country calling code and trunk code, is checked against the specified patterns. Mobile number patterns should be no longer than 20 characters, including the Country Code.
- **Use as backup device** - Turn this option on if you would like SMS to be used as a backup device. A backup device can be activated once a cardholder reports a device lost or damaged, or requests the helpdesk to disable the device temporarily.
- **OTP and Password** - Select this option when an authentication requires the cardholder to enter both a static password and one-time password.
- **SMS Centres** - Click on the link to view a list of currently configured SMS gateways. You can click on the SMSC name to edit or view the details or you can add or remove an SMSC entry by selecting the corresponding link.
- **SMS Templates** - Click on the link to Edit Default Templates for Activation During Shopping (ADS), Authentication, Activation via Authentication or Activation/Registration via MIA.
- ***Apply*** button to save changes.

SMS Centre

System Management > Authentication Management > Device Management > Edit Default Device Parameters - SMS > SMS Centres

This section is used to manage and add new SMS Centres. You can select any SMS centre to edit or delete.

- To delete an SMS Centre
  - Choose one or more SMS Centres by clicking the **Select** checkbox adjacent to the **ID**

- Click the **Delete** button.

A confirmation message will be displayed.

- To edit an SMS Centre
  - Click the *Name* hyperlink for the SMS Centre you wish to view or edit details.

The **Edit SMS Centre** page is displayed.

- To add a new SMS Centre
  - Click the *New SMS Centre

The **New SMS Centre** page is displayed.

EDIT SMS CENTRE

System Management > Authentication Management > Device Management > Edit Default Device Parameters - SMS > SMS Centres > Edit SMS Centre *displays the following fields:*

- **Device ID** is displayed and cannot be changed.
- **Name of Service provider** (mandatory).
- **Domain/IP** and **Port** - If changes are made to Domain name or IP address and Port number, they must correspond to the SMSC provider for connection to SMSC over TCP/IP.
- **System ID, System type and Password** - if changes are made, they must be specific to the parameters that are required for authentication of the client application (in this case ActiveAccess) to the SMS centre and this generally will be provided by the SMSC provider.
- **Sender's mobile number** - maximum length of 20 characters, including the Country Code. Allowed characters are A-Z, a-z, 0-9, '(', ')', '-' and space.
- **Plus (+) prefix** - Dropdown has the two options: Enabled and Disabled. Enable to add trunk code to mobile number.
- ***Apply*** button to save changes.

NEW SMS CENTRE

System Management > Authentication Management > Device Management > Edit Default Device Parameters - SMS > SMS Centres > New SMS Centre

Use the following fields to create a new SMS Centre:

- **Name** - Choose a descriptive and unique name.

If the SMS Centre is actually an MQ Server that consumes SMPP messages, the **Name** should be a unique name which will become the prefix of the required parameters in **AA_HOME/sms-jms-config.properties** for the corresponding SMSviaJMS Client. It is possible to configure as many different SMSviaJMS clients as required for ActiveAccess. For more information regarding the SMSviaJMS configuration parameters, please refer to SMS via JMS.

- **Domain/IP and Port** - Enter the Domain name or IP address and Port number provided by the SMSC provider for connection to SMSC over TCP/IP.

  ActiveAccess currently supports the following types of the SMPP gateways as SMSC and one as SMSviaSMTP:

  - **Real SMPP Compatible SMSC** - This is a real world receiver of the SMPP messages. The IP and Port of the designated SMSC need to be specified for this type. SMS maximum length is 160 ASCII characters (70 Unicode characters).
  - **SMSviaJMS Module** - This acts as an SMSC and receives SMPP messages but relays only the submit_sm messages to the MQ Server, which exclusively consumes submit_sm messages. SMS maximum length is 160 ASCII characters (70 Unicode characters).
  - **SMSviaJMS Library** - This has been embedded into ActiveAccess itself and acts as a real SMPP client but only submits the submit_sm messages to the MQ Server, which exclusively consumes submit_sm messages. **Port** can be set to any number as it does not have any usage here. SMS maximum length is 64k ASCII characters (32k Unicode characters).
  - **SMSviaSMTP Library** - This has been embedded into ActiveAccess itself and acts as an SMTP client, which builds SMS but sends them to the email addresses with a specified template in the **Domain/IP** field. **Port** can be set to any number as it does not have any usage for this type. No limitation is applied for the size of the SMS via SMTP.

Some clients have their own SMS switch, which provides all necessary information regarding SMS delivering and billing. These SMS gateways support only SMTP protocol for the incoming messages. As the ACS provides the ability to send OTP over SMTP, a template has been defined for this purpose, in the form of mailto:$DEVICE_SERIAL_NUMBER\@smtp.com.

The SMS sender module replaces the **$DEVICE_SERIAL_NUMBER** in the Domain/IP field with the registered mobile number of the cardholder and sends the OTP to a generated email account through the SMS switch.

For Example

If a cardholder has been registered with the mobile number of 614501234567, the SMS sender sends the OTP to the 614501234567@smtp.com account and the SMS switch relays it to the cardholder's mobile.

You can also define an email URL instead of an IP address for testing purposes. The email URL must start with **mailto:**, followed by the destination email address (such as **mailto:**myemail@mycompany.com). If you specify an email instead of an IP address, ActiveAccess will send the content of the SMS message to the specified email address. You must also ensure that the mail server settings are properly configured in the **_System Management** > **Settings_** page.

Alternatively, SMPPSim, an open source and free SMPP simulator from http://www.seleniumsoftware.com/, can be used for testing.

Before testing with SMS, make sure that SMS has been selected as the authentication device for the issuer and that the SMS custom pages have been loaded for the issuer.

- **System ID, System type and Password** - These are SMSC specific parameters that are required for authentication of the client application (in this case ActiveAccess) to the SMS centre and should be provided by the SMSC provider. If the SMSC does not require client authentication, leave these fields blank.
- **Sender's mobile number** - Enter the number to be used as the sender's default mobile number, for all messages sent through the selected SMSC. Maximum length is 20 characters, including the Country Code. Allowed characters are A-Z, a-z, 0-9, '(', ')', '-' and space.
- **Plus (+) prefix** - Dropdown has the two options: Enabled and Disabled. Enable to add trunk code to mobile number.
- ***Apply*** button to create the new SMS Centre.

SMS Template

Use this section to edit the default SMS templates for:

- Activation During Shopping (ADS)
- Authentication
- Activation via Authentication
- Activation/Registration via MIA.

System Management > Device Management > Edit Default Device Parameters - SMS > SMS Templates > Edit Default Templates

Use the following fields to edit an SMS Template:

- **SMS**
- **Template name** - the options are:
  - Activation During Shopping (ADS)
  - Authentication
  - Activation via Authentication
  - Activation/Registration via MIA
- **Template** - Enter the default system message. This message is sent to the cardholder when an SMS authentication is requested.

Info

Click the adjacent **Help** button for a full list of parameters. The default phrase can incorporate the following details, where appropriate:

SMS Template Parameters	Length (char)
$BatchNumber - serial number of the batch SMS sent when using device authentication of 3-D Secure	max 5
$CardExpiryDate - expiry date of the credit card	5
$CardHolderName - cardholder name as specified in the system	max 64
$CardProvider - card scheme name for the credit card	max 21
$CurrencySymbol - the currency symbol for the purchase when using device authentication over 3-D Secure	max 3
$IssuerName - Issuer's name as defined in the system	max 256 *
$MerchantCountry - 3 character country code for the Merchant's country	3

SMS Template Parameters	Length (char)
$MerchantName - Merchant's name for purchase using device authentication over	max 25 *
3-D Secure	
$MerchantURL - URL of the Merchant's website	max 2048 *
$Pan - credit card number used for device authentication over 3-D Secure	max 19
$LastFourDigitsOfPAN - last 4 digits of credit card number used for device authentication over 3-D Secure	max 4
$PurchaseCurrency - 3 character currency code for the currency of the purchase	max 3
$PurchaseDateTime - date and time of the purchase in the system	22
$PurchaseDescription - description of the purchase when using device authentication over 3-D Secure	max 125 *
$PurchaseDisplayAmount - purchase amount displayed for purchase when using device authentication over 3-D Secure	max 20
$PurchaseXID - merchant's purchase ID when using device authentication over 3-D Secure	28
$RecurringEndDate - end date for a recurring payment	10
$RecurringFrequency - recurring frequency for the purchase in days	max 4
$TokenA - the one time password. Subsequent tokens for the batch SMS can be displayed as **$TokenB**, **$TokenC**, **$TokenD**,...	max 8
$PurchaseRealAmount - indicate the transaction amount	max20

- The parameter can contain Unicode characters, but presenting Unicode characters will reduce the maximum size allowed from 160 to 70 characters.

Note

To be able to send SMS with templates in languages other than English or using symbols in the SMS Template, you must set the following system property in the application server's configuration file: `smpp.default_alphabet`.

Example

For Tomcat, set `\-Dsmpp.default_alphabet=ie.omk.smpp.util.UCS2Encoding` in the **TOMCAT_HOME/bin/catalina.bat** or **catalina.sh**.

Email

System Management > Authentication Management > Device Management > Edit Default Device Parameters - Email

Email is a virtual device natively supported by ActiveAccess to provide email OTP authentication.

The Email device can be used as a backup device.

The Email device parameters page is where the administrator can setup the system for sending OTP via email.

Use the following fields to edit email Parameters:

- **Device type** - **Email**
- **Token lifetime** - determines the validity period of email tokens in minutes (acceptable range is 0 to 10). Following the sending of an email, the token will be valid for the period specified by this option. The default lifetime of email tokens is 10 minutes. You should consider the network delay for sending email messages and give enough time for the cardholder to enter the token.
- **Token length** - determines the number of digits in the generated token (acceptable range is 6 to 10). The default size is 6 digits.
- **Maximum unsuccessful attempts to send an email** - (acceptable range is 0 to 9) if sending an OTP by email fails due to network or application errors such as connection problems to the mail server or receiving a delivery error, the system attempts to resend the email message up to the number of times specified by this option. The default value is 5. If all attempts for delivering an OTP by email fail, an error is reported back to the administration user.

- **Mail server address, Mail server port, Mail server username, Mail server password**, **Mail server protocol** and **Mail sender** - Enter the address of an outgoing SMTP mail server with a valid username and password

> **Note**
>
> The sender of the notification messages will be the main administrator user (administrator). Make sure that you have specified a correct email address for this user (use **Edit Profile** link, while logged in as the administrator).

- **Minimum wait before the updated email address can be used** (acceptable range is 0 to 9999). 0 to disable this option.
- **Use as backup device** - Turn this option on if you would like **Email** to be used as a backup device. A backup device can be activated once a cardholder reports a device lost or damaged, or requests the helpdesk to disable the device temporarily.
- **OTP and Password** - Select this option when an authentication requires the cardholder to enter both a static password and one-time password.
- **Email Templates** - Click on the link to **Edit Default Templates** for Activation During Shopping (ADS), Authentication, Activation via Authentication or Activation/Registration via MIA.
- **Send Test Email** - Click on the link to send a test email.

> **Note**
>
> The sender of the test emails will be the main administrator user (administrator). Make sure that you have specified a correct email address for this user (use **Edit Profile** link, while logged in as the administrator).

- ***Apply*** button to save changes.

Email Template

System Management > Authentication Management > Device Management > Edit Default Device Parameters - Email > Email Templates > Edit Default Templates

Use this section to edit the default email templates for:

- Activation During Shopping (ADS)
- Authentication
- Activation via Authentication

- Activation/Registration via MIA
- Subject of Activation During Shopping (ADS)
- Subject of Authentication
- Subject of Activation via Authentication
- Subject of Activation/Registration via MIA.

Use the following fields to edit an Email Template:

- **Type** - **Email** (this cannot be changed)
- **Template name** , the options are:
  - Activation During Shopping (ADS)
  - Authentication
  - Activation via Authentication
  - Activation/Registration via MIA
  - Subject of Activation During Shopping (ADS)
  - Subject of Authentication
  - Subject of Activation via Authentication
  - Subject of Activation/Registration via MIA.
- **Content type** - **Plain** or **HTML** !!! note This field is only available for templates of the email body.
- **Template** - Enter the default content for the email to be sent to the cardholder when email OTP authentication is requested.

Info

Click the adjacent **Help** button for a full list of parameters. The default phrase can incorporate the following details, where appropriate:

Email Template Parameters	Length (char)
$CardExpiryDate - expiry date of the credit card	5
$CardHolderName - cardholder name as specified in the system	max 64

Email Template Parameters	Length (char)
$CardProvider - card scheme name for the credit card	max 21
$CurrencySymbol - the currency symbol for the purchase when using device authentication over 3-D Secure	max 3
$IssuerName - issuer's name as defined in the system	max 256 *
$MerchantCountry - 3 character country code of the Merchant's country	3
$MerchantName - Merchant's name for the purchase when using device authentication over 3-D Secure	max 25 *
$MerchantURL - URL of the Merchant's website	max 2048 *
$Pan - credit card number used for device authentication over 3-D Secure	max 19
$LastFourDigitsOfPAN - last 4 digits of credit card number used for device authentication over 3-D Secure	max 4
$PurchaseCurrency - 3 character currency code for the currency of the purchase	max 3
$PurchaseDateTime - date and time of the purchase in the system	22
$PurchaseDescription - description of the purchase when using device authentication over 3-D Secure	max 125 *
$PurchaseDisplayAmount - purchase amount displayed for purchase when using device authentication over 3-D Secure	max 20
$PurchaseXID - Merchant's purchase ID when using device authentication over 3-D Secure	28
$RecurringEndDate - end date for a recurring payment	max 10
$RecurringFrequency - recurring frequency of the purchase in days	max 4
$TokenA - the one time password.	max 10
$ServicePhoneNumber - customer service phone number of the issuer	max 32

Email Template Parameters	Length (char)
$IssuerEmail - issuer's email address	max 128

* The parameter can contain Unicode characters.

OOB

System Management > Authentication Management > Device Management > Edit Default Device Parameters - OOB

OOB (Out of Band) is an API developed by the issuer to authenticate cardholders using devices that are not supported by ActiveAccess.

Use the following fields to edit OOB Device Parameters:

- **OTP and Password** - Select this option when an authentication requires the cardholder to enter a static password and complete the OOB authentication.

Backup Device

System Management > Authentication Management > Device Management > Edit Default Device Parameters - Backup Device

The backup device is a standalone backup token, which is software generated. It can be used multiple times, as configured in the Backup Device Parameters.

Use the following fields to edit Backup Device Parameters:

- **Device type** - **Backup Device**
- **Backup device lifetime** (acceptable range is 0 to 365 days)

  A value of 0 disables the device.
- **Max usage limit** - the maximum number of times the backup device can be used as (acceptable range is 0 to 9).

  A value of 0 disables the device.

- **OTP and Password** - Select this option when an authentication requires the cardholder to enter both a static password and a one-time password.
- ***Apply*** button to save changes.

VASCO

System Management > Authentication Management > Device Management > Edit Default Device Parameters - VASCO

VASCO Parameters:

- **Device type** - **VASCO**

The following manufacturer fields are available for configuration by default

- **CHECKCHALLENGE, CHKINACTDAYS, DERIVEVECTOR, DIAGLEVEL, EVENTWINDOW, GMTADJUST, HSMSLOTID, ITHRESHOLD, ITIMEWINDOW, ONLINESG, STHRESHOLD, STIMEWINDOW, STORAGEDERIVEKEY1, STORAGEDERIVEKEY2, STORAGEDERIVEKEY3, STORAGEDERIVEKEY4, STORAGEKEYID, SYNCWINDOW, TRANSPORTKEYID** and **MODE** (Response only or Challenge response) (acceptable range for field values is displayed in field hints, where appropriate).
- **OTP and Password** - Select this option when an authentication requires the cardholder to enter both a static password and one-time password.
- ***Apply*** button to save changes.

Upload File

System Management > Authentication Management > Device Management > Upload File

This page is used to enter the details of the device seed file you wish to upload and to schedule the upload date and time.

The seed file is provided by the device manufacturer.

Use the following fields to upload a file:

- **Issuer**
- **Device type**

- Click the **Choose File** / **Browse…** button, adjacent to **File name**, to locate and select a device seed file to upload.

  The **No file chosen** message will then be replaced by the **File name** of the file to be uploaded.

- **Key value** - The device manufacturer may provide a key for decrypting the seed file. Enter the key as provided by the device manufacturer.
- **Schedule Date** and **Time** when you want the uploaded data to be processed.

  Uploaded files scheduled to run in the past are set to run immediately.

  You may also leave these fields blank if you wish to process the uploaded data as soon as possible.

> **Note**
>
> The data upload may take a long time to complete depending on the file size and line speed.

- ***Apply*** button to create the upload job file.

Find Device

System Management > Authentication Management > Device Management > Find Device

Find Device can be used to search for an authentication device based on a number of criteria such as serial number, range of serial numbers, creation data and type of device.

Use the following to find a device:

- **Issuer**
- **Creation date** and **time** (dd/mm/yyyy HH:MM) or specify a date and time range for the search result by entering dates and times in the **From** and **To** fields. The date and time format is dd/mm/yyyy HH:MM. Leave the time field empty if you do not wish to limit your search for a particular time of day.
- **Device type**
- **Device Serial number** or specify a range of numbers to search within:
  - **VASCO** - device serial number, e.g. 123456789000
  - **SMS** - phone number including country code, e.g. +61123456789

- **Email** - email address, e.g. jo.citizen@domain.com

- Click ***Search*** to display device details.

Device Search Result

System Management > Authentication Management > Device Management > Find Device > Search Result

This page displays

- A list of **Devices**
- **Device ID** link to the **Device Details** page
- **Delete**, **Mark as lost**, **Mark as damaged**, **Mark as disabled** and **Back** buttons

The following fields and links are displayed for each device

- **Select** - checkbox for selecting the device to use in conjunction with the **Delete**, **Mark as lost**, **Mark as damaged** and **Mark as disabled** buttons.
- *Device ID* link to the **Device Details** page
- **Serial number** - The unique device / authentication method identifier
- **Issuer** - The issuer name to which this device belongs
- **Device type** - The type/make of the device such as VASCO, Email, SMS, etc
- **Status** - **Active**/**Lost**/**Damaged**/**Disabled**. Only an active device can be used in device authentication. If a device is reported lost, stolen, damaged or disabled, it must be flagged accordingly. A lost or damaged device can no longer be used for authentication and the cardholder must be issued with a new device.

To Delete, Mark as lost, damaged or disabled, devices in the Search Results

- Click the checkbox adjacent to the appropriate device or the checkbox in the **Select** column heading, to select all devices.

> **Warning**
>
> **Important:** The display of search results is limited to 400 records, however if you select all records, all records matching the search criteria will be affected by the action you choose to perform.

> **Warning**
>
> Performing the selected action on a large number of records may take a long time to complete and will generate the equivalent number of audit log records. Use this functionality on a large number of records diligently and only where strictly necessary.

- Click the appropriate **Delete**, **Mark as lost**, **Mark as damaged** or **Mark as disabled** button.

Device Details

System Management > Authentication Management > Device Management > Find Device > Device Details

This page is used to view details for the device selected on the **Find Device** page and to change device status if the device has been reported as lost, damaged or temporarily disabled.

The following fields and links are displayed

- **Device ID** - unique device ID
- **Issuer** - The issuer name to which this device belongs
- **Serial number** - The unique device / authentication method identifier
- **Device type** - The type / make of the device, e.g. VASCO, Email, SMS.
- **Status** - **Active/Lost/Damaged**. Only an active device can be used for authentication. If a device is reported lost, stolen or damaged, it must be flagged accordingly. A lost or damaged device can no longer be used for device authentication and the cardholder must be issued with a new device.
- **Creation date** - The date on which the device was created.
- **Reported lost/damaged on** - displays the last time a token was reported lost or damaged.
- **Device Specific Parameters** - a number of device specific parameters may be displayed for each device. These parameters are determined by the device manufacturer / authentication method and are displayed for completeness.
- **Assigned Cards** - link to a list of cards assigned to this device.
- **Activate Device** - the link appears for devices marked as lost or damaged. This allows the administrator to re-activate the device for example when the cardholder reports that the device has been found, to save the cardholder from the trouble of having to use a back up device or wait for the replacement to arrive. To activate the device, the administrator needs

to enter a valid token generated by the device to confirm that the device is actually in the possession of the cardholder again.

- **Reset device** - this option is currently supported for time-synchronous VASCO tokens. Such devices use an internal clock for generating the tokens which may gradually go out of sync with the authentication server time due to the internal clock's drift. Time synchronous devices automatically adjust this error with each authentication, as long as the time drift is within a reasonable range. The time drift on a device that has not been used for a long period of time may go outside the accepted window for automatic adjustment. In such a case, resetting the device will re-initialise the associated record and allows for a much larger window of synchronization. Before performing this action, the administrator should make sure that the cardholder's account status is enabled and should confirm that the cardholder is entering the token from a linked device by checking the device's serial number against the cardholder account. If this does not resolve the problem, the administrator should reset the token and advice the cardholder to perform another authentication. If resetting the device does not solve the problem, the device should be marked as damaged and a replacement ordered for the cardholder.

Risk Management

This section is used to set up the risk chains, which are used to define the authentication process, and the risk adapters defined in the chain. The sequence in which cardholder credentials are passed to the risk adapters is also defined in the chain. Each risk chain adapter defines a condition, actions to be taken if the condition is met or not met, a match score, together with the number of transactions which must have been performed and on how many days.

For further information about risk-based authentication and risk chains and risk adapters, refer to risk-engine-adapter.

System Management > Authentication Management > Risk Management

This page displays:

- A list of **Risk Chains** and for each Risk Chain:
  - Checkbox to **Select** it
  - **Chain ID** link to Edit Risk Chain
  - A list of **Adapters** that are enabled for the risk chain
  - Link to **Configure risk adapters** for the risk chain
- ***Delete*** button to delete selected risk chains
- Link to Add Risk Chain
- Link to Risk Adapter Management

Add / Edit Risk Chain

System Management > Authentication Management > Risk Management > Add / Edit Risk Chain

Use the following fields to complete this page:

- **Chain ID**
- **Authentication Method Score Range** - Based on the risk score (a value between 0 and 100) returned from the risk evaluation, the ranges defined in the following fields will indicate

which authentication method should be used for authenticating the cardholder for each risk score.

- **Score range for frictionless** - if the risk score falls within this range, the cardholder will be authenticated frictionlessly and authentication method will be 99.
- **Score range for frictionless with review** - if the risk score falls within this range, the cardholder will be authenticated frictionlessly and authentication method will be 97.
- **Score range for static password** - if the risk score falls within this range, the cardholder will be required to authenticate using static authentication data that has previously been assigned to them, e.g. static password.
- **Score range for device** - if the risk score falls within this range, the cardholder will be required to authenticate using an authentication device that has previously been assigned to them, e.g. SMS OTP, Email OTP, Vasco, OOB, etc. If the cardholder has multiple devices assigned, a device selection page will be displayed to them during the authentication process and they will be required to select a device from the available devices.
- **Score range for OOB** - if the risk score falls within this range, the cardholder will be required to authenticate using the authentication method utilised by the OOB service, e.g. biometrics, push notifications, etc.
- **Score range for decline** - if the risk score falls within this range, the authentication will be rejected.

Info

- The ranges defined must fully cover the range between 0 and 100
- Each range must have a begin and end value
- It is not required to have a score range for every authentication method. The score range for some authentication methods can be left blank if these authentication methods are not used by the issuer.
- The ranges can be defined in your preferred order, e.g. OOB can have a lower score range than device.
- Ranges can not overlap.

Example 1

Score range for frictionless: 0..40

Score range for static password: 41..50

Score range for device: 51..60

Score range for OOB: 61..80

Score range for decline: 81..100

Example 2

Score range for frictionless: 0..40

Score range for static password:

Score range for device: 61..100

Score range for OOB: 41..60

Score range for decline:

- ***Apply*** button to save changes
- ***Back*** button to return to the **Risk Chains** page.

Configure Risk Chain

System Management > Authentication Management > Risk Management > Configure Risk Chain

In this section, available Risk Adapters can be enabled/disabled, configured and prioritized for the corresponding risk chain.

This page displays:

- Chain ID
- A list of available **Risk Adapters** that can be configured for this Risk Chain, and for each Risk Adapter:
  - Checkbox to **Select** it. Risk Adapters can only be selected if they have been configured.
  - **Adapter ID** link to Configure Risk Adapter
  - **Move Up** and **Move Down** arrows to change the **Priority** of the risk adapter, i.e. the order in which the risk adapter is used in the risk chain

    - **Status**
        - Not configured
        - Configured
- ***Enable*** button to enable selected risk adapters
- ***Disable*** button to disable selected risk adapters
- ***Back*** button to return to the **Risk Management** page.

Configure Risk Adapter

System Management > Authentication Management > Risk Management > Configure Risk Chain > Configure Risk Adapter

This page displays:

- **Adapter ID**
- **Adapter name**

Use the following fields to complete this page:

- **Condition** which has been defined in the adapter and for each Condition:
    - **Matched behaviour** for when the Condition is matched
        - Continue
        - Finish
    - **Mismatched behaviour** for when the Condition is not matched
        - Continue
        - Finish
    - **Matched score** - the score produced when the condition is matched
    - **Condition value** - the transaction data is compared with this value to determine if it matches the condition or not.
- ***Apply*** button to save changes
- ***Back*** button to return to the **Config Risk Chain** page.

Risk Adapter Management

System Management > Authentication Management > Risk Management > Risk Adapter Management

The Risk Adapter Management page displays:

- A list of **Risk Adapters** and for each Risk Adapter:
  - Checkbox to **Select** it
  - **Adapter ID** - links to Edit Risk Adapter ID
  - **Adapter name**
  - **Risk adapter connector**
- **Delete** button to delete selected risk adapters
- Link to Register Risk Adapter
- Link to Risk Adapter Connector Management.

Register Risk Adapter

System Management > Authentication Management > Risk Management > Risk Adapter Management > Register Risk Adapter

Use the following fields to complete this page:

- **Adapter ID** can be entered by the user or generated by the system
- **Adapter name**
- Select **Risk adapter connector** from the drop down list
- ***Generate*** button to generate **Adapter ID** by the system
- ***Apply*** button to save changes
- ***Back*** button to return to the **Risk Adapter Management** page.

Edit Risk Adapter

System Management > Authentication Management > Risk Management > Risk Adapter Management > Edit Risk Adapter

Use the following fields to complete this page:

- **Adapter ID**
- **Adapter name**
- Select **Risk adapter connector** from the drop down list
- ***Apply*** button to save changes
- ***Back*** button to return to the **Risk Adapter Management** page.

Risk Adapter Connector Management

System Management > Authentication Management > Risk Management > Risk Adapter Management > Risk Adapter Connector Management

This section is used to define one or more connectors for communicating with remote risk adapters, which are called by ActiveAccess for risk-based authentication.

> **Note**
>
> To establish a secure connection with Risk Adapters, you may need CA Certificates and a keystore.

The Risk Adapter Connector Management page displays:

- A list of **Risk Adapter Connectors** and for each Risk Adapter Connector:
  - Checkbox to **Select** it
  - **Name** - links to Edit Risk Adapter Connector
  - **URL**
- **Delete** button to delete selected risk adapters
- ***Back*** button to return to the **Risk Adapter Management** page
- Link to Add Risk Adapter Connector.

Add / Edit Risk Adapter Connector

System Management > Authentication Management > Risk Management > Risk Adapter Management > Risk Adapter Connector Management > Add / Edit Risk Adapter Connector

Use the following fields to complete this page:

- **Name** of the Risk Adapter Connector
- **URL** of the Risk Adapter Connector
- **Connection timeout**
- **Read timeout**
- ***Apply*** button to save changes
- ***Back*** button to return to the **Risk Adapter Connector Management** page.

Upload Connector Encryption Key

System Management > Authentication Management > Risk Management > Risk Adapter Management > Risk Adapter Connector Management > Upload Connector Encryption Key

Use the following fields to complete this page:

- **Risk adapter connector** - choose the name of the adapter connector you want to assign an encryption key to
- **Encryption KeyStore** - click on **Browse** to locate and select an encryption key file to upload. The **No file selected** message will be replaced with the name of the file to be uploaded. The system uses the AES (128 bits) key contained in the JKS KeyStore in order to encrypt/ decrypt cardholder data that is being transferred between ActiveAccess modules and Adapter. Issuers must ensure that this AES key is used in encrypting and decrypting cardholder data at other external hosts.
- **KeyStore password** - password of the uploaded JKS KeyStore file
- ***Apply*** button to save changes
- ***Back*** button to return to the Risk Adapter Connector Management page.

OOB Management

This section is used to register and manage the OOB adapters that are used for performing Out of Band (OOB) authentication challenges. For more information about OOB adapters, refer to OOB Adapter Specification.

OOB Management

System Management > Authentication Management > OOB Management

This page displays:

- A list of **OOB Adapters** and for each adapter:
  - Checkbox to **Select** it
  - **Adapter ID** link to Edit OOB Adapter
  - **Adapter name**
  - **OOB adapter connector**
- Link to Register OOB Adapter
- Link to OOB Adapter Connector Management
- ***Delete*** button to remove selected OOB adapters.

Register / Edit OOB Adapter

System Management > Authentication Management > OOB Management > Register / Edit OOB Adapter

Use the following fields to complete this page:

- **Adapter ID** can be entered by the user or generated by the system
- **Adapter name**
- **OOB adapter connector**
- Select an **OOB adapter connector** from the drop down list.
- ***Generate*** button to generate **Adapter ID** by the system

- ***Apply*** button to save changes
- ***Back*** button to return to the **OOB Management** page.

OOB Adapter Connector Management

System Management > Authentication Management > OOB Management > OOB Adapter Connector Management

This section is used to define one or more Out of Band authentication connectors, which allow ActiveAccess to trigger the external OOB process and perform interactions with the cardholder for authentication.

This page displays:

- A list of **OOB Adapter Connectors** and for each connector:
  - Checkbox to **Select** it
  - **Name** link to Edit OOB Adapter Connector
  - **URL**
- Link to Add OOB Adapter Connector
- ***Delete*** button to remove selected OOB adapters.
- ***Back*** button to return to the **OOB Management** page.

Add / Edit OOB Adapter Connector

System Management > Authentication Management > OOB Management > OOB Adapter Connector Management > Add OOB Adapter Connector

Use the following fields to complete this page:

- **Name** of the OOB Adapter Connector
- **URL** of the OOB Adapter Connector
- **Connection timeout**
- **Read timeout**
- ***Apply*** button to save changes
- ***Back*** button to return to the **OOB Adapter Connector Management** page.

Upload Connector Encryption Key

System Management > Authentication Management > OOB Management > OOB Adapter Connector Management > Upload Connector Encryption Key

Use the following fields to complete this page:

- **OOB adapter connector** - choose the name of the adapter connector you want to assign an encryption key to
- **Encryption KeyStore** - click on **Browse** to locate and select an encryption key file to upload. The **No file selected** message will be replaced with the name of the file to be uploaded. The system uses the AES (128 bits) key contained in the JKS KeyStore in order to encrypt/ decrypt cardholder data that is being transferred between ActiveAccess modules and Adapter. Issuers must ensure that this AES key is used in encrypting and decrypting cardholder data at other external hosts.
- **KeyStore password** - password of the uploaded JKS KeyStore file
- ***Apply*** button to save changes
- ***Back*** button to return to the OOB Adapter Connector Management page.

Decoupled Authenticator Management

New page added.

This section is used to register and manage the Decoupled Authenticator adapters that are used for performing Decoupled authentication challenges. For more information about Decoupled Authenticator adapters, refer to Decoupled Authentication Adapter Specification.

Decoupled Authenticator Management

System Management > Authentication Management > Decoupled Authenticator Management

This page displays:

- A list of **Decoupled Authenticator Adapters** and for each adapter:
  - Checkbox to **Select** it
  - **Adapter ID** link to Edit Decoupled Authenticator Adapter
  - **Adapter name**
  - **Decoupled Authenticator adapter connector**
- Link to Register Decoupled Authenticator Adapter
- Link to Decoupled Authenticator Adapter Connector Management
- ***Delete*** button to remove selected Decoupled Authenticator adapters.

Register / Edit Decoupled Authenticator Adapter

System Management > Authentication Management > Decoupled Authenticator Management > Register / Edit Decoupled Authenticator Adapter

Use the following fields to complete this page:

- **Adapter ID** can be entered by the user or generated by the system
- **Adapter name**
- **Decoupled Authenticator adapter connector**
- Select an **Decoupled Authenticator server** from the drop down list.

- ***Generate*** button to generate Adapter ID by the system
- ***Apply*** button to save changes
- ***Back*** button to return to the **Decoupled Authenticator Management** page.

Decoupled Authenticator Adapter Connector Management

System Management > Authentication Management > Decoupled Authenticator Management > Decoupled Authenticator Adapter Connector Management

This section is used to define one or more Out of Band authentication connectors, which allow ActiveAccess to trigger the external Decoupled Authenticator process and perform interactions with the cardholder for authentication.

This page displays:

- A list of **Decoupled Authenticator Adapter Connectors** and for each connector:
  - Checkbox to **Select** it
  - **Name** link to Edit Decoupled Authenticator Adapter Connector
  - **URL**
- Link to Add Decoupled Authenticator Adapter Connector
- ***Delete*** button to remove selected Decoupled Authenticator adapters.
- ***Back*** button to return to the **Decoupled Authenticator Management** page.

Add / Edit Decoupled Authenticator Adapter Connector

System Management > Authentication Management > Decoupled Authenticator Management > Decoupled Authenticator Adapter Connector Management > Add Decoupled Authenticator Adapter Connector

Use the following fields to complete this page:

- **Name** of the Decoupled Authenticator Adapter Connector
- **URL** of the Decoupled Authenticator Adapter Connector
- **Connection timeout**
- **Read timeout**
- ***Apply*** button to save changes

- ***Back*** button to return to the **Decoupled Authenticator Adapter Connector Management** page.

Upload Connector Encryption Key

System Management > Authentication Management > Decoupled Authenticator Management > Decoupled Authenticator Adapter Connector Management > Upload Connector Encryption Key

Use the following fields to complete this page:

- **Decoupled Authenticator adapter connector** - choose the name of the adapter connector you want to assign an encryption key to
- **Encryption KeyStore** - click on **Browse** to locate and select an encryption key file to upload. The **No file selected** message will be replaced with the name of the file to be uploaded. The system uses the AES (128 bits) key contained in the JKS KeyStore in order to encrypt/ decrypt cardholder data that is being transferred between ActiveAccess modules and Adapter. Issuers must ensure that this AES key is used in encrypting and decrypting cardholder data at other external hosts.
- **KeyStore password** - password of the uploaded JKS KeyStore file
- ***Apply*** button to save changes
- ***Back*** button to return to the Decoupled Authenticator Adapter Connector Management page.

Public & Encryption Key Management

System Management > Public & Encryption Key Management

This section is used to provide or update the issuer's public and encryption keys. A valid public key must be defined for each issuer. The issuer system uses the issuer's public key to validate an issuer's signature. Issuers are required to sign their registration messages with a valid private key that corresponds to the public key as provided to the issuer system.

ActiveAccess uses encryption keys to encrypt cardholder data during communication between ActiveAccess and other hosts in the environment.

A KeyStore with the following details should be prepared for the encryption key that is to be uploaded, through Upload encryption key:

- **KeyStore type/format:** JCEKS
- **KeyStore provider:** SunJCE
- **Key algorithm:** AES
- **Key size:** 112 or 168 bit
- **Key name:** can be any
- **No of keys in the KeyStore:** Only one key must be populated in the KeyStore

Such KeyStores can be easily created by the Java Keytool utility using the following command:

```
keytool -genseckey -alias enckey168 -keypass 123456 -keyalg AES -keysize 168
-keystore enc-key.JKS -storepass 123456 -storetype JCEKS
```

This page displays for each key:

- **Owner**-
- **Owner Type** - type of the owner of the key, Issuer or Group
- **Certificate Information** -
- **Validity** - validity of the certificate
- **Issuer** - issuer of the certificate
- **Delete encryption key** link

- **Download public key** link
- **Download encryption key** link

Export Encryption Key

System Management > Export Encryption Key

Use the following fields to export the encryption key:

- **Encryption KeyStore** - Enter the File password.
- Click **Export**.

Upload Public Key

System Management > Upload Public Key

Use the following fields to view/ update public key details:

- **Issuer** or an **Issuer group**.A message is shown to indicate whether a public key is currently available for this item or not.
- Enter the path and filename for the **XML signing certificate**; you can use the **Choose File** / **Browse…** button.

  The system uses the public key contained in the certificate in order to validate the issuer signature when it receives messages through the registration server. Issuers must ensure that this certificate corresponds to the RSA private key, which is used in signing the registration messages.
- **Certificate information** - Displays the certificate information if one is already loaded for the selected issuer or the issuer group
- **Public Key** - Displays the public key in hexadecimal format if one is already loaded for the selected issuer or the issuer group
- ***Apply*** button to update public key information.
- ***Download*** button to save a previously uploaded certificate as a PEM encoded certificate.

Upload Encryption Key

System Management > Upload Encryption Key

Use the following fields to view / update encryption key details:

- **Issuer** or an **Issuer group**. A message is shown to indicate whether an encryption key is currently available for this item or not.
- **Choose File** button, adjacent to **Encryption KeyStore** to locate and select an encryption key file to upload.

  The **No file chosen** message or current file name will be replaced with the name of the file to be uploaded.

  The system uses the AES (128 Bits) key contained in the JKS KeyStore in order to encrypt/ decrypt cardholder data that is being transferred between ActiveAccess modules and other external hosts. Issuers must ensure that this AES key is used in encrypting and decrypting cardholder data at other external hosts.
- **KeyStore password** - File password for the Encryption KeyStore.
- **Encryption key** - Displays the key information if one is already loaded for the selected issuer or the issuer group
- ***Apply*** button to update public key information.

Exchange Configuration

System Management > Exchange Configuration

Business rules are configurable settings which provide issuers control over the customer process during the 3-D Secure transactions. The Amount Threshold rule is used to determine whether authentication can be bypassed based on an amount threshold. To cater for this requirement, it is necessary to set a threshold amount in a default currency. Where the default currency is the same as the transaction currency, this calculation is straightforward. However, where these currencies differ, it is necessary to first convert the transaction currency to an equivalent value in the default currency before calculating whether the threshold has been exceeded. In order to compare the default currency and transaction currency values where they differ, it is necessary to maintain a list of currency exchange rates.

To automate the maintenance of currency exchange values, the ACS system has been configured to automatically download an external currency exchange rate resource file. Where this list of rates is not comprehensive, and a transaction is received which is in a currency not found on the automated list, this section provides the necessary functionality to manually create currency exchange values.

The Exchange Configuration page shows a list of manually configured currency rates. The Base Currency signifies the transaction currency whilst the Target Currency signifies the currency of the issuer threshold as configured in the currency value on the Amount Threshold rule page. The Rate value is used as a multiplier, to convert an amount in the Base Currency to an equivalent amount in the Target Currency. Manual exchange rates can be edited by clicking on the *Base Currency* link.

Links are provided to *View automatic exchange rates* and *View effective exchange rates* and *Add* for adding a manual exchange rate.

This page displays:

- **Manual Exchange Rates** list
- *View automatic exchange rates* link to the **Automatic Exchange Rates** page
- *View effective exchange rates* link to the **Effective Exchange Rates** page
- *Add* link to the **Add Exchange Rate** page
- **Delete** button to allow selected exchange rates to be deleted

The following fields and links are displayed for each exchange rate:

- **Base Currency** link to the **Edit Exchange Currency** page
- **Target Currency**
- **Rate**
- **Last Update** - Shows the date and time the exchange rate was last updated.

Add Exchange Configuration

System Management > Exchange Configuration > Add > Add Exchange Rate

Use this page to add an exchange rate that is not supported by the automated currency exchange file.

Use the following fields to add a currency exchange rate:

- **Base Currency**
- **Target Currency** for the currency of the issuer threshold as configured in the currency value on the Amount Threshold rule page.
- **Rate** which should be multiplied by the Base Currency to equal the amount in the Target Currency.

> **Warning**
>
> Note that values entered here take precedence over those rates obtained by the automatic exchange rates feed when they have been updated more recently than the automatic updates.

- ***Apply*** button to save the currency exchange rate.

Edit Exchange Configuration

System Management > Exchange Configuration > Add > Edit Exchange Rate

Use this section to edit currency rates that have been manually created through the Exchange Rate section.

- On the **Exchange Configuration** page, select the *Base Currency* link.

  The **Edit Exchange Rate** page is displayed.

- The **Base Currency** and the **Target Currency** are displayed and cannot be changed.
- Enter a value for the **Rate** which should be multiplied by the Base Currency to equal the amount in the Target Currency.

> **Warning**
>
> Note that values entered here take precedence over those rates obtained by the automatic exchange rates feed when they have been updated more recently than the automatic updates.

- ***Apply*** button to save the currency exchange rate.

View Automatic Exchange Rates

Use this page to view the automatic exchange rates held in the system.

This page displays:

- **Automatic Exchange Rates** list
- **View manual exchange rates**
- **View effective exchange rates**
- **Refresh list** to update the list with the most recent exchange rates.

The following fields and links are displayed for each exchange rate:

- **Base Currency**
- **Target Currency**
- **Rate**
- **Last Update** - Shows the date and time the exchange rate was last updated.

CurrencyConvertor.properties file

You can specify the currency exchange settings by editing the **CurrencyConvertor.properties** file located in ActiveAccess' **AA_HOME** directory on the server. The following parameters are configurable in this file:

AUD_URL: Specifies the URL of the feed that provides Australian Dollar exchange rates. The default is http://www.rba.gov.au/rss/rss-cb-exchange-rates.xml.

MAX_UNSUCCESSFUL_TRY: Specifies the number of times an attempt can be made to connect to try to the exchange feed URL before giving up in the case of an error is displayed.

AUD_FILE_TYPE: Specifies the format of the currency feed for the Australian Dollar. Default is XML.

RETRY_INTERVAL: Specifies the time in seconds that the system waits before sending another request in the case of error. Default is 30.

AUD_DATE_PATTERN: Specifies the format for date and time.

UPDATE_PERIOD: Specifies how often exchange rates are updated (hours). Default is 24.

PROXY_HOST: If required specifies the proxy address to be used for connecting to the exchange feed. Default is blank.

PROXY_PORT: If required specifies the proxy port to be used for connecting to the exchange feed. Default is blank.

PROXY_USER: If required, specifies the user name to be used to connect via the proxy. Default is blank.

PROXY_PASSWORD: If required, specifies the password to be used for connecting via the proxy. Default is blank.

ActiveAccess server should be restarted for changes to take effect.

Note

Automatic Exchange Rates rely on access to the Internet or an external resource configured to retrieve these details. If no access is available, the following message is displayed "Loading effective exchange rates, please wait..."

View Effective Exchange Rate

Use this page to view the effective exchange rates.

This page displays:

- **Effective Exchange Rates** list
- **View manual exchange rates** link to the **Manual Exchange Rates** page
- **View automatic exchange rates** link to the **Automatic Exchange Rates** page

- **Refresh list** to update the list with the most recent exchange rates.

The following fields and links are displayed for each exchange rate:

- **Base Currency**
- **Target Currency**
- **Rate**
- **Last Update** - Shows the date and time the exchange rate was last updated.

> **Note**
>
> Effective Exchange Rates rely on access to the Internet or an external resource configured to retrieve these details. If no access is available, the following message is displayed "Loading effective exchange rates, please wait..."

Archive Management

System Management > Archive Management

The **Archive Management** section is used to define automatic archive settings and review the history of previous archives.

An automatic archive process can be scheduled to run at a specified time to collect records that are older than a specified date. Several archive databases can be introduced to the archive procedure but only ones which are not closed can be used for the scheduled archive procedure.

Archived databases can be chosen as the default for transaction and audit log search purposes.

Links are provided to Archive Databases, Edit Archive Settings, Archive Database Details and Archive history details pages.

This page displays:

- **Archive Settings**
- **Archive Databases** link to the **Archive Databases** page
- **Edit** link to the **Archive Settings** page

Edit Archive Settings

Use the following fields and links to edit the archive settings:

- **Automatic archive** checkbox to enable / disable automatic archiving
- **Start date** for archiving in dd/mm/yyyy format
- **Start time** for archiving in hh:mm format
- **Archive old records every**, and select **Days** or **Months** from the drop down list to specify how often records should be archived.
- **Collect records which are older than**, and select **Days** or **Months** from the drop down list to specify the age of records to archive.
- **Automatic archive purge** checkbox to enable / disable automatic archive purging
- **Purge start date** for purging archived records in dd/mm/yyyy format

- **Purge start time** for purging archived records in hh:mm format
- **Purge old archived records every**, and select **Days** or **Months** from the drop down list to specify how often records should be archived.
- **Purge archived records which are older than**, and select **Days** or **Months** from the drop down list to specify the age of records to archive.
- ***Apply*** button to save the settings

Archive Databases

This page displays:

- **Archive Databases** list
- **New Archive Database**

The following fields and links are displayed for each archive database:

- **Select** radio button to indicate which archive database to delete
- **Archive database (Archive user)**
- **Creation date** - the date that the archive database was created.
- **End date** - the date that the archive database was closed. A database is closed once a new archive database is added. Closed archive databases are not used for archiving but can still be used for Transaction/Audit Log Searches.
- **Archive user status** - indicates which archive database is the default for Transaction and Audit Log searches
- **Set as default for Search** link for selecting a different archive database to use for Transaction and Audit Log searches

New Archive Database

Use the following fields to add a new Archive Database:

- Select either **Database Link** or **Database user**
- If the **Database link** radio button is selected, enter the **Database Link**.

This must be a valid database link to an ActiveAccess archive database with the schema that has already been defined in Archive/archive_schema.sql under the ActiveAccess package.

- If the **Database user** radio button is selected, enter a valid ActiveAccess archive database, with a schema that has already been defined in Archive/archive_schema.sql under the ActiveAccess package.

> **Note**
>
> The current ActiveAccess database user should have the appropriate access rights to the archive database user objects.

- ***Apply*** button to save the new Archive Database.

> **Note**
>
> Creating a new archive database closes the current archive database and all subsequently archived records are recorded in the new archive database.
> A closed archive database can still be accessed by selecting it as the default for Transaction/Audit Log searches.

Archive Database details

This page displays:

- Links to the **Archive history details** pages

The following fields and links are displayed in Archive Database Details page

- **Archive Database** is displayed and cannot be changed
- **Creation date** is displayed and cannot be changed
- **End date -** the date that the archive database was closed. A database is closed once a new archive database is added. Closed archive databases are not used for archiving but can still be used for Transaction/Audit Log Searches.
- **Archive History** tab - lists previous archive activities with a *link* to the **Archive history details** of each archive run.
- **Purge History** tab - lists previously purged archive activities with a link to the **Purge history details** of each purge archive run.

Archive History Details

The following fields and links are displayed:

- **Archive history details** for a specified archive date and time and record age
- **Table** - name of the database table archived
- **Number of Records Archived**

Purge History Details

The following fields and links are displayed

- **Purge history details** for a specified purge date and time and record age
- **Table** - name of the database table purged
- **Number of Records Purged**

Security

System Administrators only

The **Security** section is used for setting up and maintaining digital certificates that are used for verification of connections with external parties and signing messages.

Warning

Note that server certificate related tasks that allow authentication of ActiveAccess server to external clients such as browsers and directory servers have been delegated to the ActiveAccess container. This is the application/web server which is used to run ActiveAccess server. Please consult with your application server documentation for setting up and installing SSL server certificates.

***Security** has the following sub menu options:*

- **Issuer Certificate** - for setting up and maintaining the issuers' signing certificates that are used to sign PARes messages.
- **AHS Certificate** - for setting up and maintaining client certificates used for connections to the authentication history server.
- **CAAS Certificate** - for setting up and maintaining CAAS certificates used for connections to the remote CAAS server.
- **SDK Certificate** - for setting up and maintaining SDK signing certificates that are used to sign ACSSignedContent in ARes.
- **Directory Server Certificate** - for setting up and maintaining client certificates used for connections to the Directory Server to send RReq.
- **OOB Certificate** - for setting up and maintaining client certificates used for connections to the RESTful OOB adapters.
- **Risk Certificate** - for setting up and maintaining client certificates used for connections to the RESTful RBA adapters.

- Decoupled Authenticator Certificate - for setting up and maintaining client certificates used for connections to the RESTful Decoupled Authenticator adapters.
- CA Certificate - for setting up and maintaining trusted certificates. ActiveAccess uses CA certificates to validate server certificates in outbound connections to external servers such as authentication history server.

Issuer Certificate

Security > Issuer Certificate

This section is used to setup and maintain issuers' signing certificates. Issuer certificates are used to sign PARes messages. The issuer certificates must be issued by the certificate authority designated by the 3-D Secure provider for this purpose.

The following fields and links are displayed:

- **Currently installed certificates** list
- **Create Certificate Request** for creating new certificate requests for issuers or groups
- **Install Certificate** for installation of signed certificates
- **Delete Selected Certificates** remove selected certificates.

The following fields and links are displayed for each issuer:

- **Owner**, either a group or an issuer, and links to the **Group Details** page or the **Issuer Details** page
- **Owner Type** - Shows whether the owner is a group or an issuer
- **Provider** - 3-D Secure provider of the certificate. The certificate is only used for 3-D Secure transactions, which belong to the same provider. Provider link enables certificate to be downloaded for viewing.
- **Certificate Information** - Certificate details such as Common Name (CN), Organization (O), Organization Unit (OU), Location (L), State (ST) and Country (C)
- **Validity** - Shows the validity period of the certificate
- **Status** - The status of a certificate can either be **Valid**, **Expired** or **Not signed**. You need to reapply for certificates before they expire. A certificate status is shown as not signed if the certificate is not signed by a trusted certificate authority.
- **Issuer** - The certificate authority (CA) who issued the certificate

- **Signature Algorithm** – The hash algorithm used to sign the certificate.

Create Certificate Request

Security > Issuer Certificate > Certificate Request

Use this section to create a certificate signing request (CSR) that can be sent to a designated certificate authority (CA) to obtain a signed certificate. The certificate used in signing PARes message must be signed by an appropriate CA which is designated by the scheme. You need a separately signed certificate for each supported scheme. The CSR is created in standard PKCS#10 format.

Use the following fields to create a CSR:

- Each scheme may have certain requirements regarding the format and content of CSR fields that need to be entered here. Please contact the scheme for information regarding creating a CSR. Please note that some fields may not be required by a scheme and that the following explanations are generic.
  - Select whether the CSR is for an **Issuer** or an **Issuer Group** and select the organization from the list
  - Select an authentication **Provider** (scheme) from the list
  - If the RSA Signing key is inactive, the **Alias** list is displayed and you will be required to select an **Alias**. The RSA Signing key that is created with the PCIDSS Key Retiring Utility or through **Issuers** > **Key Management** will remain inactive until a certificate request is created and signed by card schemes, then installed for the specified Alias
  - The **Key size** will be displayed once a provider and a key type (and alias, if available) have been selected. The key size is based on the size of the RSA Signing Key of the provider for each issuer.
  - Select the **Hash Algorithm** to be used to create the certificate request from the list. Defaults to **SHA1**.
  - **Common Name** - a descriptive name for the certificate, for example 'Any Bank Signing Certificate'
  - **Organization** name - for example 'Any Bank'
  - **Organizational Unit** - the name of the department within the organization to which this certificate belongs, for example 'Card Services'
  - **City** - for example 'Sydney'

- **Province** full name - for example 'New South Wales'
- **Two-letter country code** - for example AU for 'Australia.'

Install Certificate

Security > Issuer Certificate > Install Certificate

Use this section to install a certificate which is signed by the CA. The signed certificate must correspond to a previously created CSR for the same issuer and must be in standard PKCS#7 format.

Use the following fields to install a signed certificate:

- Select the appropriate radio button to indicate whether the **Issuer** or the **Issuer Group** was previously used for creating the CSR.
- Select an authentication **Provider** (scheme) from the drop sdown list. Select the provider whose CA has signed the certificate.
- If the RSA Signing key is inactive, the **Alias** list is displayed and you will be required to select an **Alias**. The RSA Signing key that is created with the PCIDSS Key Retiring Utility or through **Issuers** > **Key Management** will remain inactive until a certificate request is created and signed by card schemes, then installed for the specified Alias.
- Use the **Certificate content (file)** field to locate the PKCS#7 file that contains the signed certificate or copy and paste the signed CSR (if in base64 text format) in the **Certificate content** field.

AHS Certificate

Security > AHS Certificate

This section is used to set up and maintain SSL client certificates which are used to authenticate ActiveAccess to the authentication history server. Note that not all 3-D Secure providers may require an authentication history server. Check with the 3-D Secure provider regarding creating AHS client certificates and the designated CA for signing the certificates.

The following fields and links are displayed:

- **Currently installed certificates** list

- *Create Certificate Request* links to the **AHS Certificate Request** page for creating a new AHS client certificate request
- *Install Certificate* links to the **Install AHS Certificate** page for installation of the signed AHS client certificate
- *Delete Selected Certificates* link used with the **Select** checkbox to remove selected certificates and associated private keys
- *Import Certificate* links to the **Import AHS Certificate** page for direct installation of a signed AHS client certificate which contains a private key as well as a public key.

The following fields and links are displayed for each provider:

- **Owner** - the 3-D Secure provider and links to the **Export AHS Certificate** page. The certificate is only used for 3-D Secure transactions which belong to the same provider.
- **Certificate Information** - Certificate details such as Common Name (CN), Organization (O), Organization Unit (OU), Location (L), State (ST) and Country (C)
- **Validity** - Shows the validity period of the certificate
- **Status** - The status of a certificate can either be **Valid**, **Expired** or **Not signed**. You need to reapply for certificates before they expire. A certificate status is shown as not signed if the certificate is not signed by a trusted certificate authority.
- **Issuer** - The certificate authority (CA) that issued the certificate
- **Signature Algorithm** - The hash algorithm used to sign the certificate.

Create Certificate Request

Security > AHS Certificate > AHS Certificate Request

Use this section to create a certificate signing request (CSR) that can be sent to a designated certificate authority (CA) to obtain a signed certificate. The certificate is used in connection to the authentication history server designated by the 3-D Secure provided and must be signed by a CA approved by the respective 3-D Secure provider. The CSR is created in standard PKCS#10 format.

Use the following fields to create a CSR:

- Each scheme may have certain requirements regarding the format and content of CSR fields that need to be entered here. Please contact the scheme for information regarding creating a

CSR. Please note that some fields may not be required by a scheme and that the following explanations are generic.

- **Provider** (scheme)
- **Common Name** - a descriptive name for the certificate for example 'Any Bank AHS Client Certificate'.
- **Organization** - the name of your organization for example 'Any Bank'.
- **Organization Unit** - the name of the department within the organization to which this certificate belong for example 'Card Services'.
- **City** - for example 'Sydney'.
- **Province** - enter the state or province full name for example 'New South Wales'.
- **Two-letter country code** for example AU for 'Australia'.
- **Key size** ,defaults to **1024**.
- **Hash Algorithm** used to create the certificate request, defaults to **SHA1**.

Install AHS Certificate

Security > AHS Certificate > Install AHS Certificate

Use this section to install a certificate which is signed by the CA. The signed certificate must correspond to a previously created CSR for the same provider and must be in standard PKCS#7 format.

Use the following fields to install a signed certificate:

- **Provider** (scheme) - Select the provider whose CA has signed the certificate.
- Click the **Choose File** / **Browse…** button adjacent to **Certificate content (file)**, to locate and select the PKCS#7 file that contains the signed certificate *or* copy and paste the signed CSR (base64 text format) into the **Certificate content** text box.

Export AHS Certificate

Security > AHS Certificate > Export AHS Certificate

Use this section to export the SSL client certificate in a number of formats including PKCS#12 which allows you to export both private and public keys.

Use the following fields to export a certificate:

- **Provider** (scheme).
- **Type**, the options are:
  - **KeyStore** - to export both private and public keys
  - **Certificate** - to export the public key in DER binary encoded X509 format
  - **Certificate path** - to export the entire certificate chain in P7B format.
- If the export type selected is KeyStore, select from the **Format** list:
  - **PFX** to export in standard PKCS#12 format
  - **JKS** to export in the Java KeyStore format used by the Java Keytool and most Java-based applications.
- If the export type selected is KeyStore, enter a **File password** to protect the private key.

Import AHS Certificate

Security > AHS Certificate > Import AHS Certificate

The 3-D Secure provider may issue an SSL certificate which contains both the public and private key and is already signed. You may install this type of certificate using the import functionality provided in this section.

Use the following fields to import a certificate:

- **Provider** (scheme) .
- Select the certificate **Format.** Supported formats are **JKS** to export in the Java KeyStore format used by the Java Keytool and most Java-based applications or **PFX** to export in standard PKCS#12 format.
- Click the **Choose File** / **Browse…** button to locate and select the **File**
- Enter the **File password** which is used to protect the private key.

CAAS Certificate

Security > CAAS Certificate

This section is used to set up and maintain SSL client certificates which are used to authenticate ActiveAccess to the CAAS server. Note that the CAAS server may use mutual SSL authentication to verify the client, which in this case is ActiveAccess. Check with the CAAS server provider for more details.

The following fields and links are displayed:

- **Currently installed certificates** list
- **Create Certificate Request** for creating a new CAAS client certificate request
- **Install Certificate** for installation of the signed CAAS client certificate
- **Delete Selected Certificates** link used with the **Select** checkbox to remove selected certificates and associated private keys
- **Import Certificate** for direct installation of a signed CAAS client certificate that contains a private key as well as a public key.

The following fields and links are displayed for each provider:

- **Certificate Information** links to the **Export CAAS Certificate** page. The **Certificate Information** contains certificate details such as Common Name (CN), Organization (O), Organization Unit (OU), Location (L), State (ST) and Country (C)
- **Validity** - Shows the validity period of the certificate
- **Status** - The status of a certificate can either be **Valid**, **Expired** or **Not signed**. You need to reapply for certificates before they expire. A certificate status is shown as not signed if the certificate is not signed by a trusted certificate authority.
- **Issuer** - The certificate authority (CA) that issued the certificate
- **Signature Algorithm** - The hash algorithm used to sign the certificate.

Create Certificate Request

Security > CAAS Certificate > CAAS Certificate Request

Use this section to create a certificate signing request (CSR) that can be sent to a designated certificate authority (CA) to obtain a signed certificate. The certificate is used in connection to the authentication history server designated by the 3-D Secure provided and must be signed by a CA approved by the respective 3-D Secure provider. The CSR is created in standard PKCS#10 format.

Use the following fields to create a CSR:

- Each scheme may have certain requirements regarding the format and content of CSR fields that need to be entered here. Please contact the scheme for information regarding creating a CSR. Please note that some fields may not be required by a scheme and that the following explanations are generic.
  - **Common Name** - a descriptive name for the certificate for example 'caas-client'.
  - **Organization** - the name of your organization for example 'Internet Widgits Pty Ltd'.
  - **Organization Unit** - the name of the department within the organization to which this certificate belong for example 'Caas Services'.
  - **City** for example 'Sydney'.
  - **Province** - enter the full name of the state or province, for example 'New South Wales'.
  - **Two-letter country code**, for example AU for 'Australia'.
  - Select a **Key size** from the list. Defaults to **1024**.
  - Select the **Hash Algorithm** to be used to create the certificate request from the list. Defaults to **SHA1**.

Install CAAS Certificate

Security > CAAS Certificate > Install CAAS Certificate

Use this section to install a certificate which is signed by the CA. The signed certificate must correspond to a previously created CSR for the same provider and must be in standard PKCS#7 format.

Use the following fields to install a signed certificate:

- Click the **Choose File** / **Browse…** button adjacent to **Certificate content (file)**, to locate and select the PKCS#7 file that contains the signed certificate *or* copy and paste the signed CSR (base64 text format) into the **Certificate content** text box.

Export CAAS Certificate

Security > CAAS Certificate > Export CAAS Certificate

Use this section to export the SSL client certificate in a number of formats including PKCS#12 which allows you to export both private and public keys.

Use the following fields to export a certificate:

- Select the export **Type** from the list. The options are:
  - **KeyStore** - to export both private and public keys
  - **Certificate** - to export the public key in DER binary encoded X509 format
  - **Certificate path** - to export the entire certificate chain in P7B format.
- If the export type selected is KeyStore, select from the **Format** drop down list:
  - **PFX** to export in standard PKCS#12 format
  - **JKS** to export in the Java KeyStore format used by the Java Keytool and most Java-based applications.
- If the export type selected is KeyStore, enter a **File password** to protect the private key.

Import CAAS Certificate

Security > CAAS Certificate > Import CAAS Certificate

The CAAS server operator may issue an SSL certificate which contains both the public and private key and is already signed. You may install this type of certificate using the import functionality provided in this section.

Use the following fields to import a certificate:

- Select the certificate **Format.** Supported formats are **JKS** to export in the Java KeyStore format used by the Java Keytool and most Java-based applications or **PFX** to export in standard PKCS#12 format.
- Click the **Choose File** / **Browse…** button to locate and select the **File**
- Enter the **File password** which is used to protect the private key.

SDK Certificate

New_Section

Security > SDK Certificate

This section is used to set up and maintain SDK signing certificates which are used to sign the ACSSignedContent of ARes to the SDK via DS Server.

The following fields and links are displayed:

- **Currently installed certificates** list
- **Create Certificate Request** for creating a new SDK client certificate request
- **Install Certificate** for installation of the signed SDK client certificate
- **Delete Selected Certificates** link used with the **Select** checkbox to remove selected certificates and associated private keys.
- **Import Certificate** for direct installation of a signed SDK client certificate that contains a private key as well as a public key.

The following fields and links are displayed for each provider:

- **Certificate Information** links to the **Export SDK Certificate** page. The **Certificate Information** contains certificate details such as Common Name (CN), Organization (O), Organization Unit (OU), Location (L), State (ST) and Country (C)
- **Validity** - Shows the validity period of the certificate
- **Status** - The status of a certificate can either be **Valid**, **Expired** or **Not signed**. You need to reapply for certificates before they expire. A certificate status is shown as not signed if the certificate is not signed by a trusted certificate authority.
- **Issuer** - The certificate authority (CA) that issued the certificate
- **Signature Algorithm** - The hash algorithm used to sign the certificate.

Create Certificate Request

Security > SDK Certificate > SDK Certificate Request

Use this section to create a certificate signing request (CSR) that can be sent to a designated certificate authority (CA) to obtain a signed certificate. The certificate is used in connection to the authentication history server designated by the 3-D Secure provided and must be signed by a CA approved by the respective 3-D Secure provider. The CSR is created in standard PKCS#10 format.

Use the following fields to create a CSR:

- Each scheme may have certain requirements regarding the format and content of CSR fields that need to be entered here. Please contact the scheme for information regarding creating a

CSR. Please note that some fields may not be required by a scheme and that the following explanations are generic.

- **Common Name** - a descriptive name for the certificate for example 'sdk-client'.
- **Organization** - the name of your organization for example 'Internet Widgits Pty Ltd'.
- **Organization Unit** - the name of the department within the organization to which this certificate belong for example 'SDK Services'.
- **City** for example 'Sydney'.
- **Province** - enter the full name of the state or province, for example 'New South Wales'.
- **Two-letter country code**, for example AU for 'Australia'.
- Select a **Key size** from the list. Defaults to **1024**.
- Select the **Hash Algorithm** to be used to create the certificate request from the list. Defaults to **SHA1**.

Install SDK Certificate

Security > SDK Certificate > Install SDK Certificate

Use this section to install a certificate which is signed by the CA. The signed certificate must correspond to a previously created CSR for the same provider and must be in standard PKCS#7 format.

Use the following fields to install a signed certificate:

- Click the **Choose File** / **Browse...** button adjacent to **Certificate content (file)**, to locate and select the PKCS#7 file that contains the signed certificate *or* copy and paste the signed CSR (base64 text format) into the **Certificate content** text box.

Export SDK Certificate

Security > SDK Certificate > Export SDK Certificate

Use this section to export the SSL client certificate in a number of formats including PKCS#12 which allows you to export both private and public keys.

Use the following fields to export a certificate:

- Select the export **Type** from the list. The options are:
  - **KeyStore** - to export both private and public keys
  - **Certificate** - to export the public key in DER binary encoded X509 format
  - **Certificate path** - to export the entire certificate chain in P7B format.
- If the export type selected is KeyStore, select from the **Format** drop down list:
  - **PFX** to export in standard PKCS#12 format
  - **JKS** to export in the Java KeyStore format used by the Java Keytool and most Java-based applications.
- If the export type selected is KeyStore, enter a **File password** to protect the private key.

Import SDK Certificate

Security > SDK Certificate > Import SDK Certificate

The SDK server operator may issue an SSL certificate which contains both the public and private key and is already signed. You may install this type of certificate using the import functionality provided in this section.

Use the following fields to import a certificate:

- Select the certificate **Format.** Supported formats are **JKS** to export in the Java KeyStore format used by the Java Keytool and most Java-based applications or **PFX** to export in standard PKCS#12 format.
- Click the **Choose File** / **Browse…** button to locate and select the **File**
- Enter the **File password** which is used to protect the private key.

Directory Server Certificate

Security > Directory Server Certificate

This section is used to set up and maintain client certificates used for connections to the Directory Server to send RReq.

The following fields and links are displayed:

- **Currently installed certificates** list
- *Create Certificate Request* links to the **Directory Server Certificate Request** page for creating a new Directory Server certificate request
- *Install Certificate* links to the **Install Directory Server Certificate** page for installation of the signed Directory Server certificate
- *Delete Selected Certificates* link used with the **Select** checkbox to remove selected certificates and associated private keys
- *Import Certificate* links to the **Import Directory Server Certificate** page for direct installation of a signed Directory Server certificate which contains a private key as well as a public key.

The following fields and links are displayed for each provider:

- **Owner** - the 3-D Secure provider and links to the **Export Directory Server Certificate** page. The certificate is only used for 3-D Secure transactions which belong to the same provider.
- **Certificate Information** - Certificate details such as Common Name (CN), Organization (O), Organizational Unit (OU), Location (L), State (ST) and Country (C), Key size, Hash algorithm.
- **Validity** - Shows the validity period of the certificate
- **Status** - The status of a certificate can either be **Valid**, **Expired** or **Not signed**. You need to reapply for certificates before they expire. A certificate status is shown as not signed if the certificate is not signed by a trusted certificate authority.
- **Issuer** - The certificate authority (CA) that issued the certificate
- **Signature Algorithm** - The hash algorithm used to sign the certificate.

Create Certificate Request

Security > Directory Server Certificate > Directory Server Certificate Request

Use this section to create a certificate signing request (CSR) that can be sent to a designated certificate authority (CA) to obtain a signed certificate. The certificate is used in connection to the authentication history server designated by the 3-D Secure provided and must be signed by a CA approved by the respective 3-D Secure provider. The CSR is created in standard PKCS#10 format.

Use the following fields to create a CSR:

- Each scheme may have certain requirements regarding the format and content of CSR fields that need to be entered here. Please contact the scheme for information regarding creating a CSR. Please note that some fields may not be required by a scheme and that the following explanations are generic.
  - **Provider** (scheme)
  - **Common Name** - a descriptive name for the certificate for example 'Any Bank Directory Server Certificate'.
  - **Organization** - the name of your organization for example 'Any Bank'.
  - **Organizational Unit** - the name of the department within the organization to which this certificate belong for example 'Card Services'.
  - **City** - for example 'Sydney'.
  - **Province** - enter the state or province full name for example 'New South Wales'.
  - **Two-letter country code** for example AU for 'Australia'.
  - **Key size** ,defaults to **1024**.
  - **Hash Algorithm** used to create the certificate request, defaults to **SHA1**.

Install Directory Server Certificate

Security > Directory Server Certificate > Install Directory Server Certificate

Use this section to install a certificate which is signed by the CA. The signed certificate must correspond to a previously created CSR for the same provider and must be in standard PKCS#7 format.

Use the following fields to install a signed certificate:

- **Provider** (scheme) - Select the provider whose CA has signed the certificate.
- Click the **Choose File** / **Browse…** button adjacent to **Certificate content (file)**, to locate and select the PKCS#7 file that contains the signed certificate *or* copy and paste the signed CSR (base64 text format) into the **Certificate content** text box.

Export Directory Server Certificate

Security > Directory Server Certificate > Export Directory Server Certificate

Use this section to export the SSL client certificate in a number of formats including PKCS#12 which allows you to export both private and public keys.

Use the following fields to export a certificate:

- **Provider** (scheme).
- **Type**, the options are:
  - **KeyStore** - to export both private and public keys
  - **Certificate** - to export the public key in DER binary encoded X509 format
  - **Certificate path** - to export the entire certificate chain in P7B format.
- If the export type selected is KeyStore, select from the **Format** list:
  - **PFX** to export in standard PKCS#12 format
  - **JKS** to export in the Java KeyStore format used by the Java Keytool and most Java-based applications.
- If the export type selected is KeyStore, enter a **File password** to protect the private key.

Import Directory Server Certificate

Security > Directory Server Certificate > Import Directory Server Certificate

The 3-D Secure provider may issue an SSL certificate which contains both the public and private key and is already signed. You may install this type of certificate using the import functionality provided in this section.

Use the following fields to import a certificate:

- **Provider** (scheme) .
- Select the certificate **Type** Supported formats are **JKS** to export in the Java KeyStore format used by the Java Keytool and most Java-based applications or **PFX** to export in standard PKCS#12 format.
- Click the **Choose File** / **Browse…** button to locate and select the **File**
- Enter the **File password** which is used to protect the private key.

OOB Certificate

Security > OOB Certificate

This section is used to set up and maintain client certificates used for connections to the RESTful OOB adapters.

The following fields and links are displayed:

- **Currently installed certificates** list
- *Create Certificate Request* links to the **OOB Adapter Connector Certificate Request** page for creating a new OOB adapter connector certificate request
- *Install Certificate* links to the **Install OOB Adapter Connector Certificate** page for installation of the signed OOB adapter connector certificate
- *Delete Selected Certificates* link used with the **Select** checkbox to remove selected certificates and associated private keys
- *Import Certificate* links to the **Import OOB Adapter Connector Certificate** page for direct installation of a signed OOB adapter connector certificate which contains a private key as well as a public key.

The following fields and links are displayed for each provider:

- **OOB adapter connector** name - links to the **Export OOB Adapter Connector Certificate** page.
- **Certificate Information** - Certificate details such as Common Name (CN), Organization (O), Organizational Unit (OU), Location (L), State (ST) and Country (C)
- **Validity** - Shows the validity period of the certificate
- **Status** - The status of a certificate can either be **Valid**, **Expired** or **Not signed**. You need to reapply for certificates before they expire. A certificate status is shown as not signed if the certificate is not signed by a trusted certificate authority.
- **Issuer** - The certificate authority (CA) that issued the certificate
- **Signature Algorithm** - The hash algorithm used to sign the certificate.

Create Certificate Request

Security > OOB Adapter Connector Certificate > OOB Adapter Connector Certificate Request

Use this section to create a certificate signing request (CSR) that can be sent to a designated certificate authority (CA) to obtain a signed certificate. The certificate is used in connection to the authentication history server designated by the 3-D Secure provided and must be signed by a CA approved by the respective 3-D Secure provider. The CSR is created in standard PKCS#10 format.

Use the following fields to create a CSR:

- Each scheme may have certain requirements regarding the format and content of CSR fields that need to be entered here. Please contact the scheme for information regarding creating a CSR. Please note that some fields may not be required by a scheme and that the following explanations are generic.
    - **OOB adapter connector** - select from the list.
    - **Common Name** - a descriptive name for the certificate for example 'Any Bank OOB Adapter Connector Certificate'
    - **Organization** - the name of your organization for example 'Any Bank'
    - **Organizational Unit** - the name of the department within the organization to which this certificate belong for example 'Card Services'
    - **City** - for example 'Sydney'
    - **Province** - enter the state or province full name for example 'New South Wales'
    - **Two-letter country code** for example AU for 'Australia'
    - **Key size** ,defaults to **1024**
    - **Hash Algorithm** used to create the certificate request, defaults to **SHA1**.

Install OOB Adapter Connector Certificate

Security > OOB Adapter Connector Certificate > Install OOB Adapter Connector Certificate

Use this section to install a certificate which is signed by the CA. The signed certificate must correspond to a previously created CSR for the same provider and must be in standard PKCS#7 format.

Use the following fields to install a signed certificate:

- **OOB adapter connector** - select from the list

- Click the **Choose File** / **Browse…** button adjacent to **Certificate content (file)**, to locate and select the PKCS#7 file that contains the signed certificate *or* copy and paste the signed CSR (base64 text format) into the **Certificate content** text box.

Export OOB Adapter Connector Certificate

Security > OOB Adapter Connector Certificate > Export OOB Adapter Connector Certificate

Use this section to export the SSL client certificate in a number of formats including PKCS#12 which allows you to export both private and public keys.

Use the following fields to export a certificate:

- **OOB Adapter Connector** - select from the list.
- **Type**, the options are:
  - **KeyStore** - to export both private and public keys
  - **Certificate** - to export the public key in DER binary encoded X509 format
  - **Certificate path** - to export the entire certificate chain in P7B format.
- If the export type selected is KeyStore, select from the **Format** list:
  - **PFX** to export in standard PKCS#12 format
  - **JKS** to export in the Java KeyStore format used by the Java Keytool and most Java-based applications.
- If the export type selected is KeyStore, enter a **File password** to protect the private key.

Import OOB Adapter Connector Certificate

Security > OOB Adapter Connector Certificate > Import OOB Adapter Connector Certificate

The 3-D Secure provider may issue an SSL certificate which contains both the public and private key and is already signed. You may install this type of certificate using the import functionality provided in this section.

Use the following fields to import a certificate:

- **OOB Adapter Connector** - select from the list.

- Select the certificate **Type** Supported formats are **JKS** to export in the Java KeyStore format used by the Java Keytool and most Java-based applications or **PFX** to export in standard PKCS#12 format
- Click the **Choose File** / **Browse…** button to locate and select the **File**
- Enter the **File password** which is used to protect the private key.

Risk Certificate

Security > Risk Certificate

This section is used to set up and maintain client certificates used for connections to the RESTful RBA adapters.

The following fields and links are displayed:

- **Currently installed certificates** list
- *Create Certificate Request* links to the **Risk Adapter Connector Certificate Request** page for creating a new Risk adapter connector certificate request
- *Install Certificate* links to the **Install Risk Adapter Connector Certificate** page for installation of the signed Risk adapter connector certificate
- *Delete Selected Certificates* link used with the **Select** checkbox to remove selected certificates and associated private keys
- *Import Certificate* links to the **Import Risk Adapter Connector Certificate** page for direct installation of a signed Risk adapter connector certificate which contains a private key as well as a public key.

The following fields and links are displayed for each provider:

- **Risk Adapter Connector** name - links to the **Export Risk Adapter Connector Certificate** page
- **Certificate Information** - Certificate details such as Common Name (CN), Organization (O), Organizational Unit (OU), Location (L), State (ST) and Country (C)
- **Validity** - Shows the validity period of the certificate
- **Status** - The status of a certificate can either be **Valid**, **Expired** or **Not signed**. You need to reapply for certificates before they expire. A certificate status is shown as not signed if the certificate is not signed by a trusted certificate authority.
- **Issuer** - The certificate authority (CA) that issued the certificate

- **Signature Algorithm** - The hash algorithm used to sign the certificate.

Create Risk Adapter Connector Certificate Request

Security > Risk Certificate > Risk Adapter Connector Certificate Request

Use this section to create a certificate signing request (CSR) that can be sent to a designated certificate authority (CA) to obtain a signed certificate. The certificate is used in connection to the authentication history server designated by the 3-D Secure provided and must be signed by a CA approved by the respective 3-D Secure provider. The CSR is created in standard PKCS#10 format.

Use the following fields to create a CSR:

- Each scheme may have certain requirements regarding the format and content of CSR fields that need to be entered here. Please contact the scheme for information regarding creating a CSR. Please note that some fields may not be required by a scheme and that the following explanations are generic.
  - **Risk adapter connector** - select from the list.
  - **Common Name** - a descriptive name for the certificate for example 'Any Bank Risk Adapter Connector Certificate'
  - **Organization** - the name of your organization for example 'Any Bank'
  - **Organizational Unit** - the name of the department within the organization to which this certificate belong for example 'Card Services'
  - **City** - for example 'Sydney'
  - **Province** - enter the state or province full name for example 'New South Wales'
  - **Two-letter country code** for example AU for 'Australia'
  - **Key size** ,defaults to **1024**
  - **Hash Algorithm** used to create the certificate request, defaults to **SHA1**.

Install Risk Adapter Connector Certificate

Security > Risk Certificate > Install Risk Adapter Connector Certificate

Use this section to install a certificate which is signed by the CA. The signed certificate must correspond to a previously created CSR for the same provider and must be in standard PKCS#7 format.

Use the following fields to install a signed certificate:

- **Risk adapter connector** - select from the list
- Click the **Choose File** / **Browse…** button adjacent to **Certificate content (file)**, to locate and select the PKCS#7 file that contains the signed certificate *or* copy and paste the signed CSR (base64 text format) into the **Certificate content** text box.

Export Risk Adapter Connector Certificate

Security > Risk Certificate > Export Risk Adapter Connector Certificate

Use this section to export the SSL client certificate in a number of formats including PKCS#12 which allows you to export both private and public keys.

Use the following fields to export a certificate:

- **Risk adapter connector** - select from the list
- **Type**, the options are:
  - **KeyStore** - to export both private and public keys
  - **Certificate** - to export the public key in DER binary encoded X509 format
  - **Certificate path** - to export the entire certificate chain in P7B format.
- If the export type selected is KeyStore, select from the **Format** list:
  - **PFX** to export in standard PKCS#12 format
  - **JKS** to export in the Java KeyStore format used by the Java Keytool and most Java-based applications.
- If the export type selected is KeyStore, enter a **File password** to protect the private key.

Import Risk Adapter Connector Certificate

Security > Risk Adapter Connector Certificate > Import Risk Adapter Connector Certificate

The 3-D Secure provider may issue an SSL certificate which contains both the public and private key and is already signed. You may install this type of certificate using the import functionality provided in this section.

Use the following fields to import a certificate:

- **Risk adapter connector** - select from the list
- Select the certificate **Type** Supported formats are **JKS** to export in the Java KeyStore format used by the Java Keytool and most Java-based applications or **PFX** to export in standard PKCS#12 format
- Click the **Choose File** / **Browse…** button to locate and select the **File**
- Enter the **File password** which is used to protect the private key.

Decoupled Authenticator Certificate

Security > Decoupled Authenticator Certificate

This section is used to set up and maintain client certificates used for connections to the RESTful Decoupled Authenticator adapters.

The following fields and links are displayed:

- **Currently installed certificates** list
- *Create Certificate Request* links to the **Decoupled Authenticator Adapter Connector Certificate Request** page for creating a new Decoupled Authenticator adapter connector certificate request
- *Install Certificate* links to the **Install Decoupled Authenticator Adapter Connector Certificate** page for installation of the signed Decoupled Authenticator adapter connector certificate
- *Delete Selected Certificates* link used with the **Select** checkbox to remove selected certificates and associated private keys
- *Import Certificate* links to the **Import Decoupled Authenticator Adapter Connector Certificate** page for direct installation of a signed Decoupled Authenticator adapter connector certificate which contains a private key as well as a public key.

The following fields and links are displayed for each provider:

- **Decoupled Authenticator Adapter Connector** name - links to the **Export Decoupled Authenticator Adapter Connector Certificate** page
- **Certificate Information** - Certificate details such as Common Name (CN), Organization (O), Organizational Unit (OU), Location (L), State (ST) and Country (C)
- **Validity** - Shows the validity period of the certificate
- **Status** - The status of a certificate can either be **Valid**, **Expired** or **Not signed**. You need to reapply for certificates before they expire. A certificate status is shown as not signed if the certificate is not signed by a trusted certificate authority.
- **Issuer** - The certificate authority (CA) that issued the certificate
- **Signature Algorithm** - The hash algorithm used to sign the certificate.

Create Decoupled Authenticator Adapter Connector Certificate Request

Security > Decoupled Authenticator Certificate > Decoupled Authenticator Adapter Connector Certificate Request

Use this section to create a certificate signing request (CSR) that can be sent to a designated certificate authority (CA) to obtain a signed certificate. The certificate is used in connection to the authentication history server designated by the 3-D Secure provided and must be signed by a CA approved by the respective 3-D Secure provider. The CSR is created in standard PKCS#10 format.

Use the following fields to create a CSR:

- Each scheme may have certain requirements regarding the format and content of CSR fields that need to be entered here. Please contact the scheme for information regarding creating a CSR. Please note that some fields may not be required by a scheme and that the following explanations are generic.
  - **Decoupled Authenticator adapter connector** - select from the list.
  - **Common Name** - a descriptive name for the certificate for example 'Any Bank Decoupled Authenticator Adapter Connector Certificate'
  - **Organization** - the name of your organization for example 'Any Bank'
  - **Organizational Unit** - the name of the department within the organization to which this certificate belong for example 'Card Services'

- **City** - for example 'Sydney'
- **Province** - enter the state or province full name for example 'New South Wales'
- **Two-letter country code** for example AU for 'Australia'
- **Key size** ,defaults to **1024**
- **Hash Algorithm** used to create the certificate request, defaults to **SHA1**.

Install Decoupled Authenticator Adapter Connector Certificate

Security > Decoupled Authenticator Certificate > Install Decoupled Authenticator Adapter Connector Certificate

Use this section to install a certificate which is signed by the CA. The signed certificate must correspond to a previously created CSR for the same provider and must be in standard PKCS#7 format.

Use the following fields to install a signed certificate:

- **Decoupled Authenticator adapter connector** - select from the list
- Click the **Choose File** / **Browse…** button adjacent to **Certificate content (file)**, to locate and select the PKCS#7 file that contains the signed certificate *or* copy and paste the signed CSR (base64 text format) into the **Certificate content** text box.

Export Decoupled Authenticator Adapter Connector Certificate

Security > Decoupled Authenticator Certificate > Export Decoupled Authenticator Adapter Connector Certificate

Use this section to export the SSL client certificate in a number of formats including PKCS#12 which allows you to export both private and public keys.

Use the following fields to export a certificate:

- **Decoupled Authenticator adapter connector** - select from the list
- **Type**, the options are:
  - **KeyStore** - to export both private and public keys
  - **Certificate** - to export the public key in DER binary encoded X509 format
  - **Certificate path** - to export the entire certificate chain in P7B format.

- If the export type selected is KeyStore, select from the **Format** list:
  - **PFX** to export in standard PKCS#12 format
  - **JKS** to export in the Java KeyStore format used by the Java Keytool and most Java-based applications.
- If the export type selected is KeyStore, enter a **File password** to protect the private key.

Import Decoupled AuthenticatorDecoupled Authenticator Adapter Connector Certificate

Security > Decoupled Authenticator Adapter Connector Certificate > Import Decoupled Authenticator Adapter Connector Certificate

The 3-D Secure provider may issue an SSL certificate which contains both the public and private key and is already signed. You may install this type of certificate using the import functionality provided in this section.

Use the following fields to import a certificate:

- **Decoupled Authenticator adapter connector** - select from the list
- Select the certificate **Type** Supported formats are **JKS** to export in the Java KeyStore format used by the Java Keytool and most Java-based applications or **PFX** to export in standard PKCS#12 format
- Click the **Choose File** / **Browse…** button to locate and select the **File**
- Enter the **File password** which is used to protect the private key.

CA Certificate

Security > CA Certificate

This section is used to set up and maintain trusted certificate authority certificates. ActiveAccess uses this list in order to validate the certificate chain of installed certificates and to authenticate remote connections to external SSL enable servers such as the authentication history server.

ActiveAccess is installed with the most recent CA certificates from 3-D Secure providers. However, you may need to maintain and add new certificates they may be introduced at a later

time by the 3-D Secure provider or in order to test with non-production 3-D Secure systems that use a different CA.

The following fields and links are displayed:

- **Currently installed certificates** list
- *Import CA Certificate* links to the **Import CA Certificate** page for installation of trusted root certificates.
- *Delete Selected Certificates* link used with the **Select** checkbox to remove selected certificates.

The following fields and links are displayed for each provider:

- **Owner** - the 3-D Secure provider. Clicking on the link allows you to save the certificate in DER binary encoded X509 certificate format.
- **Type** - displays the key type
- **Certificate Information** - Certificate details such as Common Name (CN), Organization (O), Organizational Unit (OU), Location (L), State (ST) and Country (C)
- **Validity** - Shows the validity period of the certificate
- **Status** - The status of a certificate can either be **Valid**, **Expired** or **Not signed**. You need to reapply for certificates before they expire. A certificate status is shown as not signed if the certificate is not signed by a trusted certificate authority.
- **Issuer** - The certificate authority (CA) that issued the certificate.
- **Signature Algorithm** - The hash algorithm used to sign the certificate.

Import Certificate

Security > CA Certificate > Import CA Certificate

This section allows you to install additional trusted root certificates.

ActiveAccess is installed with the most recent CA certificates from 3-D Secure providers. However, you may need to maintain and add new certificates they may be introduced at a later time by the 3-D Secure provider or in order to test with non-production 3-D Secure systems that use a different CA.

Use the following fields to import a certificate:

- **Provider** - select the scheme from the list
- **Key type** - select the key type from the list
- Click the **Choose File** / **Browse…** button to locate and select the **File**. ActiveAccess supports X509 certificates in DER encoded binary or based64 encoded formats.

Servers

System Administrators only

This section is used to manage administration and access control server nodes when ActiveAccess is running in a load-balanced configuration. It is also used for setting up and maintaining authentication history servers.

When ActiveAccess is installed, the first instance of administration server and access control server are automatically recognised. However, as you expand the system by adding more administration or access control servers, for load-balancing or fail-over, you are required to introduce newly added nodes using the facility provided in this section. ActiveAccess uses these lists in order to communicate changes in the administration and options to all administration and access control server nodes.

Warning

If you do not properly introduce these servers here, the additional servers will continue to function, however they will not receive notifications when changes occur to options throughout the admin interface, which will result in system instability.

Warning

The Registration server nodes do not need to be introduced as it runs independently.

Servers has the following menu options:

- **MIA Servers** - for managing MIA Servers
- **Access Control Servers** - for managing Access Control Servers
- **Authentication History Servers** - for managing Authentication History Servers
- **Centralised Authentication and Authorisation Server** - for managing Centralised Authentication and Authorisation Servers.

MIA Servers

Servers > MIA Servers

A server entry is automatically created for the first instance of administration that you install. If you wish to install more than one server, you should first create an entry for the new server here and specify the IP address of the new instance and an arbitrary but descriptive name for the server.

This page displays:

- **MIA servers** list
- **Add Server** link
- **Delete Selected Servers** link used with the **Select** checkbox to remove selected servers.

The following fields and links are displayed for each administration server:

- **IP** link to the **Edit Server** page
- **Server Name**

Access Control Servers (ACS)

Servers > ACS Server Management

A server entry is automatically created for the first instance of ACS that you install. If you wish to install more than one server, you should first create an entry for the new server here and specify the IP address of the new instance and an arbitrary but descriptive name for the server.

This page displays:

- **Access control servers** list
- **Delete Selected ACS Servers** link used with the **Select** checkbox to remove selected servers.

The following fields and links are displayed for each administration server:

- **Server name** link to the **Edit Server** page
- **Domain name**
- **Binding IP**

- The **AHS Client** column shows whether the AHS client functionality is turned on for the ACS. If enabled the ACS will send PATransReq messages to the authentication history server.

Edit ACS Server

Servers > ACS Server Management > Server name

The following fields and links are displayed:

- **Server name**
- **Binding IP**
- **Domain name**
- **AHS client**
- Click the **Apply** button to save the changes
- Click the **Test RMI connection** button to check the status of the RMI connection.

> **Note**
>
> For Oracle WebLogic Server users, in case there is an issue in the RMI call, set `JAVA_OPTIONS="${JAVA_OPTIONS} -Dweblogic.oif.serialFilterScope=weblogic` in `setDomainEnv.cmd` or `startWebLogic.sh`

Authentication History Servers (AHS)

Servers > AHS Server Management

This section is used to define one or more authentication history servers. The authentication history server is a repository of authentication activity maintained by the 3-D Secure provider, which can be used for dispute resolution by Issuers and Acquirers. ActiveAccess sends a copy of each 3-D Secure authentication attempt to the appropriate authentication history server. Not all 3-D Secure providers support and require the transactions to be sent to an authentication history server (e.g. Visa and Mastercard require AHS but other providers do not).

This page displays:

- **Authentication history servers** list
- **Add AHS Server**
- **Delete Selected AHS Servers** link used with the **Select** checkbox to remove selected servers.

The following fields and links are displayed for each administration server:

- *URL* of the authentication history server, as provided by the 3-D Secure provider, links to the **Edit AHS Server** page.
- **ACS ID** provided by the AHS administrator for the authentication history server.
- **Login ID** provided by the AHS administrator for the authentication history server.
- **Provider**, which is the entity (e.g. Mastercard or Visa) that manages the authentication history server.

Edit AHS Server

Servers > AHS Servers> Edit AHS Server

The **Edit AHS Server** page is used to change AHS details

Fields displayed on this page:

- **Provider**

  This is the entity (e.g. Mastercard or Visa) that manages the authentication history server.

- **URL**

  This the fully qualified URL of the authentication history server as provided by the 3-D Secure provider.

- Authentication history server **ACS ID, Login ID** and **Password**

  These are provided by the AHS administrator. You will need to contact the 3-D Secure provider to obtain this information. This information is required in order to establish a successful connection to the authentication history server.

Add AHS Server

Servers > AHS Servers > AHS Server Management > Add AHS Server

The **Add AHS Server** page is used to define new AHS servers

Fields displayed on this page:

- **Provider**
- **URL**

- **ACS ID**
- **Login ID**
- **Password**

> **Info**
>
> For full information on individual fields please refer to the Edit AHS Server section of this document.

Centralised Authentication and Authorisation Servers (CAAS)

Servers > CAAS Server Management

This section is used to define one or more centralised authentication and authorisation servers. Centralised authentication and authorisation servers are remote authentication servers, which allow issuer banks to connect ActiveAccess with previously implemented remote servers that support authentication with the cardholder's existing database.

This page displays:

- **Centralised authentication and authorisation servers** list
- **Add CAAS Server**
- **Delete Selected CAAS Servers** link used with the **Select** checkbox to remove selected servers.

The following fields and links are displayed for each administration server:

- **CAAS URL**, which is the fully qualified URL of the remote authentication server (CAAS). Refer to CAAS document for further details of the URL. It links to the **Edit CAAS Server** page.
- **CAAS username**, which determines the username to access the CAAS server.

Edit CAAS Server

Servers > CAAS Servers> Edit CAAS Server

The **Edit CAAS Server** page is used to change CAAS details

Fields displayed on this page:

- **CAAS URL**

  This is the fully qualified URL of the remote authentication server (CAAS). Refer to CAAS document for further details of the URL.

- **CAAS username**

  This is the username used for accessing the CAAS server. Leave it blank if there is no username required by the CAAS authentication server.

- **CAAS password**

  This is the password associated with the CAAS username. Leave it blank if no password is required.

- **CAAS Connection timeout** in seconds (acceptable range is 60 to 9000)

  This determines the maximum amount of time the ACS, as a CAAS client, can take to complete a connection with the CAAS authentication server.

- **Maximum SMS Request** (acceptable range is 0 to 99) (0 to disable)

  This determines the maximum number of SMS requests that the ACS will attempt to initiate with the remote CAAS server. Enter 0 to disable sending SMS initialisation requests to the remote server.

- **SMS Template**

  This template is used by the remote CAAS server to send the SMS OTP via a text message.

  Use **{0}** within the template to indicate the Token/OTP.

  The following flags are available to use within the template:

  - **$LastFourDigitsOfPAN** - to indicate the last four digits of the card
  - **$MerchantName** - to indicate the merchant name for the current transaction
  - **$PurchaseRealAmount** - to indicate the transaction amount.

  > **Note**
  >
  > See SMS Template Parameters table for a full list of available parameters.

- **OOB info template**

  The template is a JSON message used by the remote CAAS server to deliver OOB required data.

The default JSON message is:

```
{
"threeDSServerTransID": "$ThreeDSServerTransID", "purchaseAmount":
"$PurchaseAmount", "purchaseCurrency": "$PurchaseCurrency",
"purchaseExponent": "$PurchaseExponent", "messageCategory":
"$MessageCategory", "deviceChannel": "$DeviceChannel", "acctNumber":
"$AcctNumber", "merchantName": "$MerchantName", "cardHolderInfo":
{"cardholderName": "$CardholderName", "email": "$Email", "homePhone": {"cc":
"$HomePhone_cc", "subscriber": HomePhone_subscriber"}, "mobilePhone": {"cc":
"$MobilePhone_cc", "subscriber": "$MobilePhone_subscriber"}, "workPhone":
{"cc": "$WorkPhone_cc", "subscriber": "$WorkPhone_subscriber"},
"shipAddrCity": "$ShipAddrCity", "shipAddrCountry": "$ShipAddrCountry",
"shipAddrLine1": "$ShipAddrLine1", "shipAddrLine2": "$ShipAddrLine2",
"shipAddrLine3": "$ShipAddrLine3", "shipAddrPostCode": "$ShipAddrPostCode",
"shipAddrState": "$ShipAddrState"}
}
```

The following flags are available to use within the template:

- `$ThreeDSServerTransID` to indicate the transaction threeDSServer Transaction ID
- `$PurchaseAmount` to indicate the transaction amount
- `$MessageCategory` to indicate the transaction message category
- `$DeviceChannel` to indicate the transaction device channel
- `$AcctNumber` to indicate the transaction account number
- `$MerchantName` to indicate the transaction merchant name
- `$CardholderName` to indicate the transaction cardholder name
- `$Email` to indicate the transaction email
- `$HomePhone_cc` to indicate the cardholder's home phone calling code in transaction
- `$HomePhone_subscriber` to indicate the cardholder's home phone number in the transaction
- `$MobilePhone_cc` to indicate the cardholder's mobile phone calling code in the transaction
- `$MobilePhone_subscriber` to indicate the cardholder's mobile phone number in the transaction
- `$WorkPhone_cc` to indicate the cardholder's work phone calling code in the transaction
- `$WorkPhone_subscriber` to indicate the cardholder's work phone number in the transaction

- `$ShipAddrCity` to indicate the transaction shipping city
- `$ShipAddrCountry` to indicate the transaction shipping country
- `$ShipAddrLine1` to indicate the transaction shipping address line 1
- `$ShipAddrLine2` to indicate the transaction shipping address line 2
- `$ShipAddrLine3` to indicate the transaction shipping address line 3
- `$ShipAddrPostCode` to indicate the transaction shipping postal code

- **Email Template**

  This template is used by the remote CAAS server to send the OTP via an email message.

  Use **{0}** within the template to indicate the Token/OTP.

  The following flags are available to use within the template:

  - **$LastFourDigitsOfPAN** - to indicate the last four digits of the card number
  - **$MerchantName** - to indicate the merchant name for the current transaction
  - **$PurchaseRealAmount** - to indicate the transaction amount
  - **$ServicePhoneNumber** - to indicate the issuer's customer service phone number
  - **$IssuerEmail** - to indicate the issuer's email address

  > **Note**
  >
  > See Email Template Parameters table for a full list of available parameters.

- **Email Subject Template**

  This template is used by the remote CAAS server for the Subject to be used for the OTP via email message.

  The flags described in Email Template above can be used for the Subject template.

- Select the **Use Proxy** checkbox if the ACS is to connect to the remote CAAS server via a proxy and complete the following:
  - **Proxy host**, which determines the proxy's IP address or domain name.
  - **Proxy port**, which determines the proxy's port.
  - **Proxy username**, which determines the proxy's username, if required.
  - **Proxy password** associated with the Proxy username, if required.
  - ***Apply*** button to save updated settings.

- Click the **Check CAAS Status** link to verify that the CAAS server can be reached by the current remote authentication settings.

  The **Check CAAS Status** will be displayed, which shows the current status of the remote authentication server and is used to indicate the remote authentication server is running or not. * Click the **Retry** button to re-test the remote authentication server status.

- Click the **Close** button to close this page.

Add CAAS Server

Servers > CAAS Servers > CAAS Server Management > Add CAAS Server

The **Add CAAS Server** page is used to define new CAAS servers

Fields displayed on this page:

- **CAAS URL**
- **CAAS username**
- **CAAS password**
- **CAAS Connection timeout**
- **Maximum SMS request**
- **SMS template**
- **Email template**
- **Email subject template**
- **Use proxy**
- **Proxy host**
- **Proxy port**
- **Proxy username**
- **Proxy password**

> **Info**
>
> For full information on individual fields, please refer to the Edit CAAS Server section.

Utilities

System Administrators only

This section is used to load, run and manage add-ins from within the ActiveAccess administration. Utilities can be assigned to, and run on behalf of, an Issuer or Issuer Group.

Utilities *has the following sub menu options:*

- **Utilities** - used for managing add-in utilities
- **Upload Utility** - used to upload add-in utilities on behalf of one or a number of Issuer or Issuer Groups.

The first **Utilities** page is **Utilities Search Result**.

Note

Utilities shown may vary.

Utilities

Utilities > Utilities Search Results

The Utilities section is used for viewing the details and availability of utilities in the system, and for managing and running selected utilities.

Note

Utilities shown may vary.

This page displays:

- Utilities List
- **Delete** button to allow selected utilities to be deleted

The following fields and links are displayed for each utility:

- **Select** checkbox
- **Name**
- **File Name**
- **Version**
- **Issuer**
- **Group**
- **Creation time**
- **Run** link - links to the **first** page of the selected Utility.

Upload Utility

Utilities > Upload Utility

Use this page to upload utilities to the system or on behalf of one or a number of Issuers or Issuer Groups.

Use the following fields to upload a utility:

- **Select the appropriate Issuer** or **Group** to upload the utility for.
- **File name** - click the Choose File button to locate and select the utility file to upload.
- ***Apply*** to upload the selected utility.

Key Retiring Utility

> *Previously AA85 - PCIDSS Key Retiring Utility.pdf*

This PCIDSS Key Retiring utility retires specified encryption keys and regenerates new encryption keys. Related table columns are then re-encrypted by new keys. The utility allows for the automatic retiring of old keys and regeneration of new ones, while keys manually created by HSM administrators can also be introduced as new keys.

In the case of automatic key retiring, the utility can be run for selected general MIA / ACS settings encryption keys or issuer / groups based on key type and provider. Replacement with manually created keys can be carried out for general MIA / ACS settings encryption keys or issuers by entering the new key alias of keys created by the HSM administrator.

Uploading the Utility

A System Administrator will be responsible for uploading the utility through the MIA (**Utilities > Upload Utility**).

To upload the utility

- There is no need to select an Issuer or Group to upload this utility.
- Browse to locate and select the **File name** (PCIDSSKeyRetiringUtility.war).
- Click the **Apply** button to upload the utility.

  The utility will be listed in the MIA utilities section (**Utilities > Utilities**) **PCIDSS Key Retiring Utility**.

Running the Utility

This utility makes changes in the HSM keystore and re-encrypts cardholder data and configuration settings in the database. Therefore, a full backup of the HSM keystore and ActiveAccess database should be taken before running this utility. If any archive database has been configured for automatic archiving, a backup of its database should be taken as well.

> **Note**
>
> Automatic archiving and purging in **System Management > Archive Management** must be disabled before running the utility. During the utility run, all ActiveAccess modules must stop receiving requests from the outside world.

Utility List

To run the utility

- Go to the MIA utilities section (**Utilities > Utilities**)
- Click the *Run* link adjacent to the **PCIDSS Key Retiring Utility**.

  The **PCIDSS Key Retiring Utility** screen is displayed prompting users to select which issuer, group or general encryption keys to run the utility for.

Retiring keys automatically

To customise the key retiring process

- Select **Retire old keys and generate new ones automatically**

To retire keys automatically

- Select the **General encryption keys** radio button
  - Select the **MIA settings encryption key** checkbox
  - Select the **ACS settings encryption key** checkbox
- Select the **Issuer** radio button
  - Select the **Issuer** from the drop down list
- Click the **Prepare** button

> **Note**
>
> The process for retiring General/Data Encryption Keys occurs in two stages: Preparation and Finalization. For stage one of the process, the **Prepare** button will be available.

Retiring keys using manually created keys

Select the following field to customise the key retiring process

- Select **Retire old encryption keys and use the keys which have been created by HSM administrator**

Use the following fields for retiring keys using manually created keys

- Select the **General encryption keys** radio button
  - Select the **MIA settings encryption key** checkbox and enter the created key in the **New key alias** field
  - Select the **ACS settings encryption key** checkbox and enter the created key in the **New key alias** field
- Select the **Issuer** radio button
  - Select the **Issuer** from the drop down list
  - Enter the created key in the **New key alias** field
- Click the **Prepare** button

Results

When the process is complete, the Results will be available for immediate display. For more details of the utility process you can check **AA_HOME/mia_log.log**.

Encryption Key - Preparation Failure

If there is a failure within any of the steps of the preparation process, the utility stops and logs the details of the issue for the administrator's reference.

Encryption Key - Failed Resume/Rollback

If the encryption key retiring fails in the preparation stage, the process can be resumed from the latest status once the issue is resolved or all the changes can be undone using the Rollback option. When a process has a failed status, new processes cannot be started until the current process is successfully resumed or rolled back.

Encryption Key - Preparation Success

During the encryption key retiring process, a new encryption key is generated and a temporary column is added to the specified table for every column that keeps encrypted data. The data from the main column is decrypted using the old key, then encrypted using the new key and stored in a temp column.

Encryption Key - Finalization Re-encrypt/Finalize/Rollback

Once the preparation stage of the encryption key retiring process is completed successfully, the process can be finalized.

If any new data has been created after the completion of the preparation stage, the encryption process can be redone using the **Re-encrypt** option.

Alternatively, all the changes made during the preparation stage can be undone using the **Rollback** option.

The **Finalize** option completes the encryption key retiring process. In the Finalization process, once all the required columns are re-encrypted, the main column is dropped and the **temp** column is renamed to the name of the main column. In the final step, all the required constraints and indexes are created for the main column.

When the MIA settings encryption key is automatically or manually retired and replaced with a new one

If there are any other instances of MIA, Registration servers, rather than the current server, in the environment, replace the **DBOWNERPASSWORD** and **DBPASSWORD** values with their plain values in the **AA_HOME/activeaccess.properties** file, then add the following properties to it and restart:

HSMENCALIAS=MIA_DB_DESEDE_NEW (where MIA_DB_DESEDE_NEW is the new MIA settings encryption key alias in HSM)

PLAIN_TEXT=

When the ACS settings encryption key is automatically or manually retired and replaced with a new one

If there is any other instance of ACS, rather than the current server, in the environment, replace the **DBOWNERPASSWORD** and **DBPASSWORD** values with their plain values in the **AA_HOME/activeaccess.properties** file, then add the following properties to it and restart:

HSMENCALIAS=AA_Administration_NEW (where AA_Administration_NEW is the new ACS settings encryption key alias in HSM)

PLAIN_TEXT=

When the Issuer's data encryption key is automatically or manually retired and replaced with a new one

The current notification report files of the selected issuer are no longer valid and will be re-collected in the next run of the specified job in the Registration server.

If there is any other instance of Registration server, rather than the current instance, in the environment, add the following property into the **AA_HOME/activeaccess.properties** file:

NOTIFICATION_REPORT_REGEN_ISSUERIDS=1234567890 (where 1234567890 is the Issuer ID)

If property **NOTIFICATION_REPORT_REGEN_ISSUERIDS** already exists, modify its value by appending the Issuer ID to the end and restart.

Encryption Key - Archive

Following the successful finalization of the encryption key retiring process, if archiving is configured on the system, the encryption key of the archive database must also be retired and replaced using the **Re-encrypt Archive** option.

Migrate to Data Key Utility

New page added.

This utility retires the current encryption keys and uses the new data encryption keys which have been generated during the installation process. The fields that are currently kept encrypted with the HSM encryption keys will be decrypted by the current HSM keys, and re-encrypted using the new data encryption keys.

Uploading the Utility

A System Administrator will be responsible for uploading the utility through MIA (**Utilities > Upload Utility**).

To upload the utility:

- Do **not** select an Issuer or Group to upload this utility.
- Browse to locate and select the **File name** (MigrateToDataKeyUtility.war).
- Click the **Apply** button to upload the utility.

  The utility will be listed in the MIA utilities section (**Utilities > Utilities**): **Encryption Key Migration Utility**.

Running the Utility

To run the utility:

- Go to the MIA utilities section (**Utilities > Utilities**)
- Click the *Run* link adjacent to the **Encryption Key Migration Utility**'s Creation Time.

  The **Encryption Key Migration Utility** screen is displayed prompting users to run the key retiring process on the main database.

Warning

After the completion of the utility run, the current notification report files will no longer be valid and will be re-collected in the next run of the specified job in the Registration server.

If there are other instances of ActiveAccess servers, in addition to the current instance, move `NOTIFICATION_REPORT_REGEN_ISSUERIDS` property into **AA_HOME/activeaccess.properties** and restart ActiveAccess.

Results

When the process is complete, the results will be available for immediate display. For further details about the utility run process, please refer to **AA_HOME/mia_log.log**.

If the process failed, please check **AA_HOME/logs/mia_log.log** regarding the cause of failure. The process can be resumed once the issue is resolved by clicking **Resume** button.

The utility will automatically retire the old encryption keys and activate the new data keys.

Note

- If archive users exist, click **Run on archive** to complete the re-encryption process.

Issuers

System Administrators and **Issuer Administrators** only

This section is used to set issuer specific settings; maintain and upload card details; create and maintain custom pages and manage keys.
When a new issuer is created all issuer settings are set to default.
Issuers has the following menu options:

***Issuers** has the following sub menu options:*

- Settings
- Upload Registration Files
- Registration Requests
- Custom Pages
- Key Management

The first **Issuers** page is **Settings**.

Settings

This section is used to set up and maintain issuer system settings for displaying PARes, providing proof of authentication attempts to merchants, maximum unsuccessful authentication attempts permitted for cardholders and the automatic unlock lag time.

Note

Settings are different for **Local** and **Remote** Issuers.

Local Issuer Settings

These settings are available if the Issuer's **Authentication server** has been set to **Local** in Issuer Details.

Issuer > Settings

Use the following fields to manage Local Settings:

- Select an **Issuer** from the drop down list, to display its settings.
  This field is not displayed if the user is assigned to a single issuer.
- **Issuer ID** cannot be changed.
- **BINs** - displays a list of BINs currently assigned to the selected issuer. The BIN list can only be changed by a user with System Admin access level from **System Management > Issuer Management** section.
- **Maximum authentication attempts** allows the administrator to setup an upper limit for the number of successful authentications that can be performed by each user (acceptable range is 0 to 999) in a specified period of time (acceptable range is 0 to 24 hours). This is particularly useful when the issuer is being charged per transaction for each authentication and it makes sense to set an upper limit for the financial liability.

  This option is disabled by default which means the number of successful authentications that can be performed by the user is not limited.

  Once a set limit is reached, further authentication attempts fail at the UEReq/UERes level with status code 'U' and reason code '5' (maximum number of transactions exceeded).
- **Maximum unsuccessful attempts** that will be permitted for unsuccessful authentication or enrolment attempts by cardholders (acceptable range is 0 to 9).

  The default value is **3**, which means that 4 unsuccessful authentication or registration attempts with a card will result in the card being locked to avoid further access for security reasons. An issuer may change to any other value to comply with their internal policy.

  > **Warning**
  >
  > Setting this field to 0 disables the automatic locking mechanism and is not recommended.

- **Maximum interaction** is used to set a maximum number of cardholder interactions as determined by the selected Challenge Flows and security requirements to allow an appropriate number of cardholder retries without going beyond the pre-set maximum

(acceptable range is 0 to 10). When the limit is reached, the transaction fails but the card will not be locked.

- **Automatic unlock** time in minutes (acceptable range is 0 to 1440). A currently locked card can be automatically unlocked after the amount of time specified here has passed.

  This may help to reduce helpdesk calls if set properly.

  The default value is **0**, which implies that this field is disabled and as such all locked accounts have to be manually unlocked by helpdesk staff.

- Specify the cardholder **Password policy** using the following:
  - **Minimum password length** between 1 and 128 chars (typically 6)
  - **Maximum password length** between 1 and 128 chars (typically 16)
  - **Minimum password digit**, the minimum number of numerical characters the password must contain. The default value is 0, which disables this field.
  - **Minimum password capital letter**, the minimum number of capital letters the password must contain (typically 1). The default value is 0, which disables this field.

> **Note**
>
> The sum total of the numbers entered for **Minimum password digit** and **Minimum password capital letter** must be less than or equal to the **Minimum password length**.

- **Time zone**

  This allows administrators to set an individual time zone for the specified issuer.

  The default time zone is set when the application is installed and is displayed for reference, on the menu bar, from where it can be modified at any time, as and when appropriate. Modification of the Time zone on the menu bar *does not* change the Time zone for the Issuer in the Issuer Settings.

> **Note**
>
> If you modify the Time zone in the menu bar it will persist for the current session only. It will revert to the Time zone entered in the Issuer settings, the next time you login.
>
> All search parameters for transactions, audit logs and reports (daily, monthly and annual) will be based on the Time zone specified on the menu bar at the time of the search.

> **Warning**
>
> **IMPORTANT:** If the time zone in **Issuers > Settings** is changed, it will impact the data displayed for issuer reports (daily, monthly and annual). When attempting to change the time zone, a warning message is displayed with the following options:
>
> - **Continue and delete report data** - reports will not be available for the selected issuer until the next overnight report run, which will use the new time zone.
>
> **NOTE** :If auto archive is enabled, archived data will no longer be collected and previous report data will be lost.
>
> - **Continue and keep report data** - existing report data will be inaccurate due to the time change. Accurate reports will not be available until the next overnight report run, which will use the new time zone.
> - **Cancel** - time zone will not be changed.

- **Language selection during authentication**

  This allows administrators to enable or disable the language selection page displayed to cardholders during the authentication process of 3-D Secure 1 and the challenge process of 3-D Secure 2 authentications.

- **Name on card verification**

  This allows administrators to enable or disable cardholder name When it is Enabled then ACS will not compare the received cardholder name in request with saved value.

- A link is provided to Provider Settings.

Provider Settings

There are a number of settings that can be specified per authentication scheme. You should set these parameters in accordance with the recommendation of the 3-D Secure authority of each scheme.

Issuer > Settings > Provider Settings

Use the following fields to view / edit Providers settings:

- Select an **Issuer** from the drop down list
- Select a **Provider ID** from the drop down list
- Select **Enabled** or **Disabled** from the **Activation during shopping** drop down list to enable or disable the cardholder registration during the shopping process.

  Enabling this option allows an issuer to dynamically enrol the cardholders while they are shopping at a 3-D Secure enabled merchant site. The activation during shopping process only applies to those cardholders who have been pre-registered by their issuer in the system.

- Select **Enabled** or **Disabled** from the **Proof of authentication attempt** drop down list to enable to disable providing authentication attempt guarantee to merchants.

  This option applies to SafeKey, SecureCode, ProtectBuy, J/Secure and Verified by Visa in 3-D Secure version 1.0.2 and later. An issuer may choose to provide proof of authentication attempts for non-enrolled cardholders, when an authentication is requested by the merchant. Proof of attempt processing provides guarantee of funds transfer to the merchant. This may shift the liability to the issuer despite the fact the cardholder was not enrolled and could not be authenticated. Proof of attempt is an incentive for the merchants to implement 3-D Secure.

- Specify the value for **Maximum ADS proof of attempts** (acceptable range is 0 to 9). The option limits the number of times a user is allowed to opt-out of ADS processes and still receive proof of authentication attempt status code. Once the limit is reached, cancelling ADS will result in PARes status='N' to be returned to the merchant and it is likely that cardholder transaction will not be authorised by the merchant. Set this option to 0, if you wish to grant unlimited authentication attempts to cardholders.
- Specify the value for **PAReq freshness period** in minutes (acceptable range is 0 to 60). The default value 0, which effectively disables this option.

  An ACS may receive duplicate PAReq messages due to cardholder actions (for example, if the cardholder clicks the **Back** or **Refresh** buttons during the authentication process). In order to provide good customer service, and minimise cardholder confusion, the 3-D Secure protocol recommends that receipt of a duplicate PAReq within a reasonable time should not be treated as an error. This is called the **PAReq freshness period**. According to the 3-D Secure bulletin of July 12, 2004, the recommended period should be between 10 and 15 minutes.

> **Warning**
>
> ActiveAccess sends a PARes with status code 'U' and iReqCode 56, if a duplicate PAReq is received outside the period specified by this parameter.

> **Warning**
>
> The ADS and attempt process for Visa, American Express, Diners Club International and JCB is the same but different for Mastercard. Mastercard does not currently recognise attempt processing in the sense defined by Visa specification and does not provide authentication guarantee and liability shift if the cardholder is not enrolled. However, Mastercard still requires a PARes with status 'A' to be sent when the cardholder cancels ADS up to the limit defined by the issuer. For more information refer to Visa 3-D Secure standard and Mastercard SecureCode specification.

- *Mastercard SecureCode only*: Select **Mastercard SecureCode** or **Mastercard Identity Check** from the **Authentication type** drop down list.
- *American Express SafeKey only*: Specify the value for **Maximum forgot password attempts** (acceptable range is 0 to 9, default is 2 as specified in the SafeKey Issuer Implementation Guide). The option limits the number of times a user is allowed to enter an incorrect SafeKey before the card is locked. Once the limit is reached, it will result in PARes status='N' to be returned to the merchant and the cardholder transaction may not be authorised by the merchant.
- Select **A (Attempted)** or **N (Not approved)** from the **Unsupported device PARes status** drop down list.

  This option specifies the PARes to be used for unsupported devices.
- Specify any **Browser Unsupported devices** in the text box

  This is for specifying browsers / devices for which authentication is not supported in browser mode. It can also be used to quickly remove support if, for example, a security issue has been reported for a particular browser.

  Format of the input is JSON array.

  Example

  ```
  [{"user-agent":"Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:69.0) Gecko/20100101 Firefox/69.0"}]
  ```

- Specify any **App Unsupported devices** in the text box

  This is for specifying browsers / devices for which authentication is not supported in app mode. It can also be used to quickly remove support if, for example, a security issue has been reported for a particular browser.

  Format of the input is JSON array.

Example

[{"DV":"1.0","DD":{"C001":"Android","C002":"HTC One_M8","C004":"5.0.1","C005":"en_US","C006":"Eastern Standard Time","C007":"06797903-fb61-41ed-94c2-4d2b74e27d18","C009":"John's Android Device",....},"DPNA":{"C010":"RE01","C011":"RE03"},"SW":["SW01","SW04"]}, {"DV":"1.0","DD":{"C001":"iOS","C002":"iPhone 5c","C003":" iPhone OS","C004":"9.2","C005":"en-US","C006":"GMT-6","C009":"John's iPhone",....}," DPNA":{"C010":"RE01","C011":"RE03"},"SW":"SW01","SW04"]}, {"DV":"1.0","DD":{"C001":"Windows","C002":"NOKIA RM-984_1006","C003":"WindowPhone","C004":"10.0.10586.11","C005":"en-US","C006":"(UTC-06:00) Central Time (US & Canada)","C007":"1bbd95da4520a6dfe7b94480d69f3cbb","C008":"1280x720","C009":"My Phone",....},"DPNA":"C010":"RE02","C011":"RE03"},"SW":["SW01","SW04"]}
]

- Set the **Challenge Mandated Indicator**

  The ACS decides based on the ACS Challenge Mandated Indicator, the 3DS Requestor Challenge Indicator, and the ACS Rendering Type whether to perform the requested challenge.

- **Cardholder info for non-exempt authentication** - Text provided by the ACS to the cardholder during a frictionless transaction that was not authenticated by the ACS.

  It is optional for issuers to provide information to the cardholder.

Example

"Additional authentication is needed for this transaction, please contact (Issuer Name) at xxx-xxx-xxxx."

- **ACS Reference Number** - A unique reference number provided by EMVCo to ActiveAccess.
- **ACS Operator ID** - An ACS identifier assigned by the Directory Server. Each Directory Server can provide a unique ID to each ACS on an individual basis.
- **Broad Info** - Unstructured information sent between the 3DS Server, the Directory Server and the ACS.
- *J/Secure only*: **Display attempt time** - The duration of displaying an attempt page for JCB cards only in case of attempt returning status. A value of **0** indicates that no attempt page should be shown.

Remote Issuer Settings

These settings are available if the Issuer's **Authentication server** has been set to **Remote (CAAS)** in Issuer Details.

Issuers > Settings

Use the following fields to view/ edit Remote Settings:

- **Issuer** This field is not displayed if the user is assigned to a single issuer.
- **Issuer ID** cannot be changed
- **Maximum interaction** is used to set a maximum number of cardholder interactions as determined by the selected Challenge Flows and security requirements to allow an appropriate number of cardholder retries without going beyond the pre-set maximum (acceptable range is 0 to 10). When the limit is reached, the transaction fails but the card will not be locked.
- **Time zone**

  This allows administrators to set an individual time zone for the specified issuer.

  The default time zone is set when the application is installed and is displayed for reference, on the menu bar, from where it can be modified at any time, as and when appropriate. Modification of the Time zone on the menu bar *does not* change the Time zone for the Issuer in the Issuer Settings.

> **Note**
>
> If you modify the Time zone in the menu bar it will persist for the current session only. It will revert to the Time zone entered in the Issuer settings, the next time you login. All search parameters for transactions, audit logs and reports (daily, monthly and annual) will be based on the Time zone specified on the menu bar at the time of the search.

> **Warning**
>
> **IMPORTANT:** If the time zone in **Issuers > Settings** is changed, it will impact the data displayed for issuer reports (daily, monthly and annual). When attempting to change the time zone, a warning message is displayed with the following options:
>
> - **Continue and delete report** data - reports will not be available for the selected issuer until the next overnight report run, which will use the new time zone.
>
> **NOTE**- If auto archive is enabled, archived data will no longer be collected and previous report data will be lost.
>
> - **Continue and keep report data** - existing report data will be inaccurate due to the time change. Accurate reports will not be available until the next overnight report run, which will use the new time zone.
> - **Cancel** - time zone will not be changed.

Authentication Scheme Settings

There are a number of settings that can be specified per authentication scheme including activation during shopping, attempt processing and PAReq freshness period. You should set these parameters in accordance with the recommendation of the 3-D Secure authority of each scheme.

- Select **Enabled** or **Disabled** from the **Use ACS local settings** drop down list.

  Enabling this option allows issuer settings to be set locally in the ACS instead of remotely on the CAAS side.

> **Note**
>
> If **Disabled**, refer to Remote Messaging Specification for further information on setting these parameters.

*If **Enabled**:*

- Select **Enabled** or **Disabled** from the **Activation during shopping** drop down list to enable or disable the cardholder registration during the shopping process.

  Enabling this option allows an issuer to dynamically enrol the cardholders while they are shopping at a 3-D Secure enabled merchant site. The activation during shopping process only applies to those cardholders who have been pre-registered by their issuer in the system.

- Select **Enabled** or **Disabled** from the **Proof of authentication attempt** drop down list to enable to disable providing authentication attempt guarantee to merchants.

  This option applies to SafeKey, SecureCode, ProtectBuy, J/Secure and Verified by Visa in 3-D Secure version 1.0.2 and later. An issuer may choose to provide proof of authentication

attempts for non-enrolled cardholders, when an authentication is requested by the merchant. Proof of attempt processing provides guarantee of funds transfer to the merchant. This may shift the liability to the issuer despite the fact the cardholder was not enrolled and could not be authenticated. Proof of attempt is an incentive for the merchants to implement 3-D Secure.

- Specify the value for **Maximum ADS proof of attempts** (acceptable range is 0 to 9). The option limits the number of times a user is allowed to opt-out of ADS processes and still receive proof of authentication attempt status code. Once the limit is reached, cancelling ADS will result in PARes status='N' to be returned to the merchant and it is likely that cardholder transaction will not be authorised by the merchant. Set this option to 0, if you wish to grant unlimited authentication attempts to cardholders.
- Specify the value for **PAReq freshness period** in minutes (acceptable range is 0 to 60). The default value 0, which effectively disables this option.

  An ACS may receive duplicate PAReq messages due to cardholder actions (for example, if the cardholder clicks the **Back** or **Refresh** buttons during the authentication process). In order to provide good customer service, and minimise cardholder confusion, the 3-D Secure protocol recommends that receipt of a duplicate PAReq within a reasonable time should not be treated as an error. This is called the **PAReq freshness period**. According to the 3-D Secure bulletin of July 12, 2004, the recommended period should be between 10 and 15 minutes.

> **Warning**
>
> ActiveAccess sends a PARes with status code 'U' and iReqCode 56, if a duplicate PAReq is received outside the period specified by this parameter.

> **Warning**
>
> The ADS and attempt process for Visa, American Express, Diners Club International and JCB is the same but different for Mastercard. Mastercard does not currently recognise attempt processing in the sense defined by Visa specification and does not provide authentication guarantee and liability shift if the cardholder is not enrolled. However, Mastercard still requires a PARes with status 'A' to be sent when the cardholder cancels ADS up to the limit defined by the issuer. For more information refer to Visa 3-D Secure standard and Mastercard SecureCode specification.

- **Mastercard SecureCode only**: Select **Mastercard SecureCode** or **Mastercard Identity Check** from the **Authentication type** drop down list.

- **American Express SafeKey only**: Specify the value for **Maximum forgot password attempts** (acceptable range is 0 to 9, default is 2 as specified in the SafeKey Issuer Implementation Guide). The option limits the number of times a user is allowed to enter an incorrect SafeKey before the card is locked. Once the limit is reached, it will result in PARes status='N' to be returned to the merchant and the cardholder transaction may not be authorised by the merchant.
- Select **A (Attempted)** or **N (Not approved)** from the **Unsupported device PARes status** drop down list.

  This option specifies the PARes to be used for unsupported devices.
- Specify any **Unsupported devices** in the text box

  This is for specifying browsers / devices for which authentication is not supported. It can also be used to quickly remove support if, for example, a security issue has been reported for a particular browser.

  Separate multiple browsers / devices using commas (,). This setting is not case sensitive.

Upload Registration Files

Issuers > Upload Registration Files

The **Upload Registration Files** section is used to upload user registration or card registration messages for bulk registration or pre-registration of cardholders and users. Files can be uploaded for an individual Issuer or for an Issuer Group.

> **Info**
>
> Please see the **Users** section for information on registering and managing individual card and user accounts.

The main page shows a report on recently performed file uploads and their status.

You can schedule uploading a user or card registration file using the *Upload File* link and, view details using the *Job* number link, and schedule or cancel scheduled uploads using the *Edit* or *Cancel* links.

> **Note**
>
> This page will not be available for remote issuers.

Uploading XML files that contain SMS devices

When uploading XML files that contain SMS devices, note that if the national trunk prefix of the mobile number has been entered (`0` or `1`), these digits will automatically be removed from the start of the mobile number by ActiveAccess.

Use the following fields and links for managing card uploads:

- The first available **Issuer** is displayed by default. If you are assigned to an issuer group, select All or an **Issuer** from the drop down list and click the adjacent **Refresh** button.

  A list of the selected issuer's card files and their status is displayed.
- Select **All** or the type of registration message from the **Message Type** drop down list.

  A list of the selected issuer's card files and their status is displayed.
- The default report is for the last 10 days, but you can specify an upload **Date** range for the search result by entering dates in the **From** and **To** fields using dd/mm/yyyy format and clicking the **Refresh** button.
- Click the *Upload File* link (which is only displayed when an Issuer is selected) to schedule a file upload job for the selected issuer.

  The **Upload File** page is displayed.

The following details are displayed for each uploaded file for the selected issuer:

- **Job Number** - this number is defined by the system and links to the **Job Details** page, which provides full details for the job and details of any error messages or warning conditions
- **Issuer**- name of the issuer that owns the card upload job
- **Group** - name of the issuer group that owns the card upload job
- **Message Type**- the type of registration message either card registration or user registration
- **File Name** - the name of the file uploaded
- **Started** - the date and time the file upload started
- **Finished** - the date and time the file upload finished
- **Attempts** - the number of times the file upload was attempted
- **Status** - the upload status shows the current status for data upload, which can be one of the following:
  - Completed

- Completed with warnings
- Processing
- Failed
- Scheduled
- Cancelled

- **Edit** link - displayed for **Scheduled** uploads.

  This links to the **Edit Upload Details** page for updating the scheduled date and time.

- **Cancel** link - displayed for **Scheduled** uploads.

  The administrator may cancel a scheduled upload, by clicking the *Cancel* link, but cannot cancel one which is in progress.

Upload File

Issuers > Upload Registration Files > Upload File

This page is used to enter the details of the card file you wish to upload and to schedule the upload date and time.

Use the following fields to upload a file:

- Choose the appropriate radio button and select an **Issuer** or an **Issuer group** from the drop down list.
- Select the type of registration message either **Card Registration** Or **User Registration** from the **Message Type** drop down list
- Click the **Choose File** / **Browse…** button adjacent to **File name**, to locate and select a registration file to upload.

  The **No file chosen** message will then be replaced by the File name of the file to be uploaded.

- Enter **Schedule Date** and **Time** when you want the uploaded data to be processed.

  Uploaded files scheduled to run in the past are set to run immediately.

  You may also leave these fields blank if you wish to process the uploaded data as soon as possible.

> **Note**
>
> The data upload may take a long time to complete depending on the file size and line speed.

Job Details

Issuers > Upload Registration Files > Job Details

This page provides job details and a link to the registration request details via the Message ID link. It also provides information on any error conditions that prevented the upload from being processed successfully.

The fields displayed are:

- **Issuer** name
- **Job number**
- **Message ID** link to **Request details**
- **Message type** - Card Registration or User Registration
- **Uploaded** date and time
- **File name**
- When the upload was **Started** and **Finished**
- Number of **Attempts** before the upload was finished
- **Status** of the job
- **Error message**
- **Error Details**
- **Warnings**

Edit Uploaded File

Issuers > Upload Registration Files > Edit Uploaded File

This page is used to update the scheduled processing time by specifying a new **Date** and **Time**.

Use the following fields to edit the file's scheduled upload:

- **Issuer** - cannot be changed

- **Message type** - cannot be changed
- **File name** - cannot be changed.

  To upload a different file you must first cancel this upload using the **Cancel** link on the **Upload Registration File** page and then select the **Upload File** link.
- **Date** using dd/mm/yyyy format
- **Time** using hh:mm format
- **Apply** button to save.

Registration Requests

Issuers > Registration Requests Search

The **Registration Requests** section is used to view requests for user registration or card registration messages

You can view registration request details using the **REG ID** link.

> **Note**
>
> This page will not be available for remote issuers.

Use the following fields to find a registration request:

- Select from the **Issuer** drop down list to limit the results to the specified issuer **OR**
- Select from the **Group** drop down list to limit the results to the specified issuer group.
- The **Request ID** is the identifier entered in the registration message by the issuer. Enter all of the **Request ID** to search.
- The default **Creation date** range is for the last 10 days, but you can specify a date and time range (inclusive) in the **From** and **To** fields. The date and time format is dd/mm/yyyy HH:MM. Leave the time field empty if you do not wish to limit your search for a particular time of day.
- The default **Completion date** range is for the last 10 days, but you can specify a date and time range (inclusive) in the **From** and **To** fields. The date and time format is dd/mm/yyyy HH:MM. Leave the time field empty if you do not wish to limit your search for a particular time of day.

- Select the **Status** of the registration requests from the drop down list. The options are:
  - All (default)
  - Completed
  - Completed with warnings
  - Failed
  - Processing
- Click **Search** to display registration request details

A list of the selected issuer or issuer group's registered requests and their progress is displayed.

Issuers > Registration Requests displays:

The following details are displayed for each registration request

- **REG ID** - this number is defined by the system and links to the **Request Details** page, which provides full details for the request job and details of any error messages or warning conditions
- **Issuer** - name of the issuer who owns the registration request
- **Group** - name of the issuer group who owns the registration request
- **Creation date** - the date the request was created
- **Completion date** - the date the request was completed
- **Request ID** - the Request ID associated with the request message
- **Progress** - the status of the request; Completed, Completed with warnings, Failed or Processing.

Custom Pages

Issuers > Custom Pages

This section is used to upload, store and manage issuer branded pages. Branded pages are displayed to cardholders during authentication processes.

Each issuer is assigned a separate space and a separate URL for their authentication pages. Issuers can modify the XSL files of the pages to include the issuer's logo and customised text.

This ensures that cardholders will always be presented with their own issuer branded pages during the authentication process.

Page customisation

When customising the pages, note that the main text that appears on the page should not exceed 350 characters. In addition to this, the maximum length of `card number` is 19 characters, `amount` is 48 characters, and `merchant name` is 40 characters.

Uploading tip

For ease of upload, you can zip the files first and upload them all at once.

Sample custom pages

A set of sample custom pages, with *Any Bank* branding, is available in the ActiveAccess installation package: ActiveAccess/data/custompage/issuer

The naming convention for 3DS1 authentication pages is as follows:

Page	Filename
J/Secure authentication	auth_jcb_index.xsl
SecureCode authentication	auth_spa_index.xsl
VbV authentication	auth_vbv_index.xsl
SafeKey authentication	auth_sk_index.xsl
ProtectBuy authentication	auth_dc_index.xsl
Two-factor device authentication	dev_index.xsl

Other resources can be uploaded to the issuer space such as help files and graphics, etc.

To avoid any run time problems or security risk, only trained personnel can upload branded pages. As such, the option to upload custom pages is available at the **system administration** level only.

Issuer administrators have read-only access to this function, which can be used to download custom pages and branded material.

> **Note**
>
> The issuer system limits issuer space to a flat file structure (i.e. all files are created at the same directory level.

You can upload new pages using the **Upload File** link and **Delete** or **Download** pages.

Use the following fields and links for managing the custom pages:

- Select an **Issuer** from the drop down list of available issuers and click the **Refresh** button.

  A list of the issuer's custom pages is displayed.

  Or

- Select a **Group** from the drop down list of available groups and click the **Refresh** button.

  A list of the group's custom pages is displayed.

- **Upload File** link to upload a new file for the selected issuer or issuer group.

  The **Upload File** page is displayed.

- **Download Selected** link, used in conjunction with the **Select** checkbox to download one or multiple custom pages, for the selected issuer or issuer group.
- **Delete Selected** link, used in conjunction with the **Select** checkbox to delete one or multiple custom pages, for the selected issuer or issuer group

The following custom page details are displayed for the selected issuer:

- **File Name**
- **Size** - size of file in bytes
- **Date** - date and time of upload
- **Delete** link to delete the page
- **Download** link to download the page

Upload File

Issuers > Custom Pages > Upload File

This page is used to enter the name and location of the custom page you wish to upload

Use the following fields to upload a file:

- Select the **Issuer** for which you are uploading the custom pages from the drop down list
- Alternatively a **Group** can be selected from the drop down list. Selecting a group allows the administrator to roll out an update to all the issuers that are a direct member of the group or a member of a group owned by the selected group.

> **Note**
>
> **Important:** Care should be taken when rolling out an update to a group as it will overwrite the corresponding files on all the member issuers. Issuers may have configuration, graphics or text files specific to their own brand. You should not upload a generic package that overwrites these issuer branded pages through this facility without carefully checking first.

- Click the **Choose File** / **Browse…** button, adjacent to **File name**, to locate and select a custom page file to upload.

  The **No file chosen** message will then be replaced with the name of the file to be uploaded
- Click the **Apply** button to upload the file.

  File upload confirmation is displayed and if uploaded pages support rules, a link is provided to allow issuer to use rules.

Key Management

The system creates a number of cryptographic keys for each issuer in order to protect sensitive and confidential information. These keys are securely stored in the ActiveAccess database by utilising a Master Key on a hardware security module (HSM) to encrypt/decrypt these keys.

This section lists keys used by the issuer and the history of any changes. The list of keys is retrieved by the MIA instance, which is currently being accessed by the user. It is the responsibility of the system administrator to keep all HSM instances synchronised at all times.

This section also allows the administrator to retire the current Signing RSA or CAVV validation keys and create new ones. Card and general encryption keys cannot be retired and replaced using this interface as a process to decrypt previously encrypted fields with an old key and re-encrypt them using a new key is required. GPayments has developed a PCIDSS Key Retiring Utility for this purpose.

Use the following fields and links for viewing keys:

- The first available **Issuer** is displayed by default. Select a different **Issuer** from the drop down lists of available issuers and click the adjacent **Refresh** button.

  A list of the selected issuer's current keys is displayed.
- Select the **Group** radio button to select an **Issuer Group** from the drop down lists of available issuer groups, and click the adjacent **Refresh** button.

  A list of the selected issuer group's current keys is displayed.
- Select the **General keys** radio button to view the list of the general encryption keys that are used to encrypt general critical settings and configuration parameters.
- *New Key* link to the New Key page
- *Import Key* link to the Import Key page.

This page displays for each key;

- *Alias* link to the **Key Details** page
- **Delete** button to allow unused keys to be deleted

> **Note**
>
> Delete is only available for non-encryption keys

- **Export** button to allow exporting of keys, and links to **Export Data Key**

> **Note**
>
> Export is only available for HMAC and CAVV keys

*The following key details are displayed for the selected **Issuer** or **Group**:*

- **Provider**
- **Algorithm**
- **Type**
- **Alias**
- **Creation time** - date and time of upload
- **Status**

- **KeyStore type** - possible values: Data, HSM

*The following key details are displayed for the **General keys**:*

- **Algorithm**
- **Type**
- **Alias**
- **Creation time** - date and time of upload
- **Status**
- **KeyStore type** - possible values: Data, HSM

New Key

Issuers > Key Management > New Key

The **New Key** section is used to retire the current Signing RSA, CAVV validation, or HMAC keys, and replace them with a new key.

Alternatively, the **PCIDSS Key Retiring Utility** provided in the ActiveAccess installation package allows for the automatic retiring of old keys and re-generation of new ones. Refer to Key Retiring Utility for further details.

Use the following fields and links for generation of new keys:

- **Issuer** or **Group**
- **Type**
- **Provider**

> **Note**
>
> **SecureCode HMAC generation key** and **SecureCode HMAC 256 generation key** are only available for the Mastercard provider.

- **Algorithm** is displayed and cannot be changed
- **Old alias** is displayed and cannot be changed
- **Old key size** is displayed and cannot be changed
- **New alias status** is displayed and cannot be changed

- **New alias**
- If the key **Type** is **Signing RSA key**, select a **Key size** from the drop down list. Defaults to **1024**.
- Click the **Generate new key** button.

> **Info**
>
> - In order to use the newly created **Signing RSA key**, you need to create a certificate request using this key and have the certificate signed. Then the signed certificate must be installed for the key to be used in the next transaction.
> - In order to use the newly created **CAVV key**, you must activate it in Key Details before it can be used for the next transaction.
> - In order to use the newly created **HMAC key**, you must activate it in Key Details before it can be used for the next transaction.

Import Key

Issuers > Key Management > Import Key

The **Import Key** section is used for importing CAVV and HMAC keys.

Use the following fields and links for importing keys:

- **Issuer** or **Group**
- **Type**
- **Provider**

> **Note**
>
> **SecureCode HMAC generation key** and **SecureCode HMAC 256 generation key** are only available for the Mastercard provider.

- **Algorithm** is displayed and cannot be changed
- **Old alias** is displayed and cannot be changed
- **New alias status** is displayed and cannot be changed
- **Key value**
- Click the **Import key** button.

Key Details

Issuers > Key Management > Key Details

The **Key Details** section is used to list the history of the changes for the specified key.

The following key details are displayed for the selected alias:

- **Alias**
- **Algorithm**
- **Creation time** - date and time of upload
- **Expiration time** - date and time the key will expire
- **Status**
  - **Active** - the key is being used by the system
  - **Inactive** - the key needs to be activated through a pre-defined process
  - **Expired** - the key has been retired and will no longer be used by the system
- **KeyStore type** - the key has been retired and will no longer be used by the system
  - **Data** - the key is stored in the database
  - **HSM** - the key is stored in the HSM
- Click the **Activate** button to activate inactive keys
- Click the **Delete** button to delete unused keys.

Export Data Key

Issuers > Key Management > Export

The **Export Data Key** section is used to export HMAC and CAVV keys.

The following fields are displayed to view / edit:

- **Issuer** or **Group**
- **Provider**
- **Type**
- **Alias** - the alias of the key to be exported

- **KEK alias** (Key Encryption Key alias) - the alias of an encryption key stored in the HSM, which is required for encrypting the key that is to be exported.
- Click the **Export** button to save the key.

Rules

System Administrators and **Issuer Addministrators**
Access can also be granted to **Business Admin** and **Helpdesk** users at System Admin level.
Whether these users have *read only* or *full* access is determined by their Admins settings.

This section is used to set up and manage business rules and the sequence in which they are applied for Issuers that have rules functionality enabled.

The rules that can be applied are determined by whether the authentication server is remote or local. Authentication exemption rules and settings can be applied for local and remote authentication servers and registration enforcement rules can be applied for local authentication servers.

You can set up and maintain Issuer authentication exemption business rule settings for the local or remote ACS (CAAS). These business rules are configurable settings, which provide Issuers with control over the customer process during 3-D Secure transactions as described in the table below. Rules can be configured using 3-D Secure transaction parameters such as Transaction Amount, Merchant ID, Merchant Name, Acquirer BIN or Merchant Country. The sequence in which the rules are applied can be defined by setting Priority values for each rule.

Rule Management has the following tabs:

- **Registration** - allows two pre-defined rules to be used for checking authentication requests processed or transparently authenticated by a local authentication server. The Rules are:
  - **Amount Threshold**
  - **Merchant Blacklist**

  These rules can be enabled, disabled and have their order of priority changed.
- **Authentication**
  - **Soft Launch List**
  - **Merchant Whitelist**
  - **Merchant Watchlist**

- **Location Watchlist**
- **Domestic & International Transaction Amount Threshold**

These rules can be enabled, disabled and have their order of priority changed.

- **Settings** - used to set up transaction number and / or amount thresholds to be used for determining if authentication is to be initiated or bypassed. Also used to define the authentication response (PARes) to be sent for any transactions where authentication is bypassed.

Rules can be set to apply by default and not just for exceptions.

Registration

Rules > Rule Management > Registration

This section is used to set up and maintain settings for business rules that force customer registration. These business rules are configurable settings which provide issuers control over the customer process during 3-D Secure transactions.

These rules can be configured using 3-D Secure transaction parameters such as the Transaction Amount, Merchant ID, Merchant Name, Acquirer BIN or Merchant Country.

The sequence in which the rules are applied can be changed by setting a Priority for each rule.

Rule	Description	Parameters	Behaviour	
Amount Threshold	A threshold that determines if pre-registered cardholders are to be forced to opt-in to Activation During Shopping.	**Transaction amount** (AUD or converted to AUD)	*Greater than or equal to threshold:* No Opt-Out option available on the ADS page, only cancel	*Less than threshold:* Opt-Out option will be available on the ADS page *Opt-Out* - sets transaction status to 'A'

Rule	Description	Parameters	Behaviour	
Merchant Blacklist	Merchant criteria that initiate the authentication.	**Merchant** Merchant ID Merchant name Acquirer BIN Merchant country	*On list:* No Opt-Out option available on the ADS page, only cancel *Cancel* - sets Transaction Status to 'N'.	*Not on list:* Opt-Out option will be available on the ADS page *Opt-Out* - sets transaction status to 'A'

Use the following fields to view / edit the Rules:

- If the required **Issuer** or an **Issuer Group** is not displayed, select it from the appropriate drop down list.
- Click the **Refresh** button.

> **Note**
>
> These fields are not displayed if the user is assigned to a single issuer.

The following fields and links are displayed for each rule:

- **Select** checkbox to be used in conjunction with the **Enable** and **Disable** buttons.
- *Rule Name* link to the **Rule Details** page
- **Priority** sequence in which the rules are applied, can be changed by clicking **Move Up** or **Move Down**
- **Status** - Enabled, Disabled or Not Configured

Use the following steps to enable or disable a rule:

- Choose one or more rules by clicking the **Select** checkbox adjacent to the **Rule Name**
- Click the **Enable** or **Disable** button as appropriate.

  A confirmation message will be displayed.

Amount Threshold

Rules > Rule Management > Registration > Amount Threshold

The Amount Threshold rule allows an issuer to encourage pre-registered cardholders to register for 3-D Secure based on the value of the transaction.

When this rule is enabled, the cardholder is pre-registered and when they purchase from a 3-D Secure enabled merchant, if the value of the transaction is equal to or exceeds the threshold amount, the cardholder will be presented with the standard ADS page but will not be given the opportunity to Opt-Out. Instead, the cardholder will only be able to Cancel the transaction and the authentication result that will be returned to the merchant will be, **N**.

If the transaction is in a currency other than the one selected for the rule, a conversion will be made to convert the value of the transaction into the currency of the rule, so that the converted value can be compared with the threshold to determine the appropriate action.

> **Note**
>
> If currency conversion is not available, the default will be for registration to be enforced.

The following fields and links are displayed:

- **Select** checkbox for selecting one or more sets of rules.
- **BIN** links to the **Edit Amount Threshold** page.
- **Delete** button used in conjunction with selected rules.
- Links for **Manage Exchange Rates** and **Add**.

To manually create currency exchange values:

- Click the **Manage Exchange Rates** link.

  The **Manage Exchange Rate** page is displayed.

To add Amount Threshold rules:

Rules > Force Registration > Amount Threshold > Add

- Click the **Add** link.

  The **Add Amount Threshold** page is displayed.

> **Note**
>
> The *Add* link is disabled when a rule has been set for All BINs of the selected issuer.

To delete a rule:

- Select the checkbox adjacent to the BIN and click the **Delete** button.

  Confirmation of the deletion is displayed.

To set transaction amount thresholds:

- **Issuer** name is displayed and cannot be changed.
- Select the **BIN** to be used for the threshold from the drop down list.
- Enter an **Amount** for the threshold.
- Select the default currency to be used for the domestic threshold from the **Currency** drop down list.
- Click the **Apply** button to save changes.

  A confirmation message will be displayed.

Merchant Blacklist

The Merchant Blacklist rule allows an issuer to encourage pre-registered cardholders to register for 3-D Secure based on merchant related transaction parameters. If the attributes of the business rule match the merchant related transaction parameters, the cardholder will not be allowed to Opt-Out of the Activation During Shopping process and will only be able to cancel the transaction. Issuers can use any combination of Merchant ID, Merchant Name, Acquirer BIN and Merchant Country to create a rule to encourage cardholders to register.

If the cardholder cancels the transaction, the authentication result returned to the merchant will be an **N**.

When checking the Merchant details from a transaction against the Merchant Blacklist, the Merchant is considered to be a match if any of the Merchant details (Merchant ID, Merchant name, Acquirer BIN or Merchant country) from the PAReq match the corresponding details on the Blacklist.

You can set a maximum number of rules for the Merchant Blacklist, import a file of Merchant details or add individual Merchant details to the list, delete Merchant details and edit Merchant details.

Merchant Blacklist Details

Rules > Rule Management > Registration > Merchant Blacklist > Merchant Blacklist Search Results

You can add individual Merchant details or import a file of Merchant details to the list or select any Merchant details to delete or edit.

The following fields and links are displayed:

- **Maximum no of rules in blacklist** (default is 50)
- **Select** checkbox for selecting one or more sets of Merchant details
- **Delete** button used in conjunction with selected Merchant details
- **Merchant ID**, **Merchant Name**, **Acquirer BIN** and **Merchant Country** links to the **Edit Merchant Blacklist** page
- Links for **Add** and **Import**

To change the number of rules permitted in the blacklist:

- Enter the **Maximum no of rules in blacklist**.
- Click the **Apply** button.

To add individual Merchant details:

- Click the **Add** link.

  The **Add Merchant Blacklist** page is displayed.

To import a file of Merchant details:

- Click the **Import** link.

  The **Import Merchant Blacklist** page is displayed.

> **Note**
>
> The supported file formats for uploading rule configurations are .csv and .xml. The following is an example of a sample .xml file:

```
<MerchantBlacklist>
    <Merchant>
```

```
        <Id>1</Id>
        <Name>hi</Name>
        <AcquirerBIN>412345</AcquirerBIN>
        <CountryCode>024</CountryCode>
    </Merchant>
    <Merchant>
        <Id>12134567890</Id>
        <Name>hi</Name>
        <AcquirerBIN>412345</AcquirerBIN>
    </Merchant>
</MerchantBlacklist>
```

To delete Merchant details:

- Select the checkbox adjacent to the Merchant ID and click the **Delete** button.

  Confirmation of the deletion is displayed.

Edit Merchant Blacklist

Rules > Rule Management > Registration > Merchant Blacklist > Edit Merchant Blacklist

To edit the Merchant Blacklist:

- **Issuer** name is displayed and cannot be changed.
- Edit any of the Merchant fields:
  - **Merchant ID**
  - **Merchant Name**
  - **Acquirer BIN**
  - **Merchant Country**.
- Click the **Apply** button. A confirmation message will be displayed.

When checking whether a transaction matches the Merchant Blacklist, the system will compare the Merchant ID, Merchant Name, Acquirer BIN and Merchant Country from the PAReq, with the values entered.

- Click the **Back** button to return to the Merchant Blacklist.

Import Merchant Blacklist

Rules > Rule Management > Registration > Merchant Blacklist > Import Merchant Blacklist

This page is used to import a file of merchant Blacklist rules to add to the Merchant Blacklist. The format of the file should be CSV, with each record incorporating values for any of the required Merchant ID, Merchant Name, Acquirer BIN and Merchant Country fields necessary to define the rule.

> **Note**
>
> Merchant country is defined by a country code and name and uses the format:
> < ISO 3166-1 numeric code>/< ISO 3166-1 country name>
>
> For example:
>
> 008/Albania
>
> 024/Angola
>
> 036/Australia

To import a file of merchant watchlist rules:

- **Issuer** name is displayed and cannot be changed.
- Click the **Choose File** / **Browse** button adjacent to **File name**, to locate and select a file to import.

  The **No file chosen** message will be replaced with the name of the file to be imported.

  > **Note**
  >
  > A maximum of 1000 records can be imported at one time.

- Click the **Apply** button.

  A confirmation message will be displayed.

Authentication

Rules > Rule Management > Authentication

Rule	Description	Parameters	*If on list*	*If not on list*
Soft Launch List	Primary Account Numbers that will not be allowed to bypass the authentication procedure.	**Cardholder:** Primary Account Number	Other business rules applied in the specified sequence.	Cardholder transparently authenticated. Transaction Status set to 'Y' or 'A', as configured under the Settings tab.
Merchant Whitelist	Merchant criteria that allow the authentication procedure to be bypassed.	**Merchant:** Merchant ID Merchant name Acquirer BIN Merchant country	Cardholder transparently authenticated. Transaction Status set to 'Y' or 'A', as configured under the Settings tab.	Other business rules applied in the specified sequence.
Merchant Watchlist	Merchant criteria that initiate the authentication procedure.	**Merchant:** Merchant ID Merchant name Acquirer BIN Merchant country	Initiates authentication procedure.	Other business rules applied in the specified sequence.
Location Watchlist	Merchant countries that initiate the authentication procedure	**Merchant country:** Country / Currency code	Initiates authentication procedure.	Other business rules applied in the specified sequence.
Domestic & International Transaction Amount Threshold	A threshold that determines if the authentication procedure is to be initiated or bypassed. Domestic threshold used for the set default transaction purchase currency, otherwise, International threshold used.	**Transaction amount**	*Greater than or equal to threshold:* Initiates authentication procedure.	*Less than threshold or currency not in file:* Cardholder transparently authenticated. Transaction Status set to 'Y' or 'A', as configured under the Settings tab.

Rule	Description	Parameters	*If on list*	*If not on list*
Stand-In Transaction	A threshold that determines if the authentication procedure is to be bypassed when the remote authentication server (CAAS) cannot be contacted or is not responding at the Verify Registration (first message to CAAS) stage.	**Transaction amount** (AUD or converted to AUD)	*Greater than or equal to Stand-in threshold:* Unable to authenticate Transaction Status set to 'U'	*Less than Stand-In threshold:* Cardholder transparently authenticated. Transaction Status set to 'Y' or 'A', as configured under the Settings tab.

- Click the **Rules** tab, if it is not already selected.

Use the following fields to view / edit the Rules:

- If the required **Issuer** or an **Issuer Group** is not displayed, select it from the appropriate drop down list.
- Click the **Refresh** button.

> **Note**
>
> These fields are not displayed if the user is assigned to a single issuer.

The following fields and links are displayed for each rule:

- **Select** checkbox to be used in conjunction with the **Enable** and **Disable** buttons.
- **Rule Name** link to the **Rule Details** page
- Once two or more rules are enabled, the **Priority** sequence in which the rules are applied can be changed by clicking **Move Up** or **Move Down**. Once Priority is customised, click **Reset to Default**, to reset the sequence in which the rules are applied back to the system default.
- **Status** - Enabled, Disabled or Not Configured

To enable or disable a rule:

- Choose one or more rules by clicking the **Select** checkbox adjacent to the **Rule Name**

- Click the ***Enable*** or ***Disable*** button as appropriate.
  A confirmation message will be displayed.

Soft Launch List Rule

If the cardholder's Primary Account Number is on the Soft Launch List, the authentication procedure will not be bypassed, and the other business rules will be applied in the specified sequence.

> **Warning**
>
> If the cardholder is not on the list, the cardholder will be considered to be transparently authenticated and transaction status will be set to **Y** or **A**, as configured under the **Settings** tab.

You can search for PANs in the Soft Launch List, import a file of PANs or add an individual PAN to the list and edit a PAN.

The first page in this section is **Search Soft Launch List**.

Search Soft Launch List

Rules > Rule Management > Soft Launch List > Search Soft Launch List

This page displays:

- **Issuer** name = cannot be changed.
- **Search** button to search for cardholder PANs already in the Soft Launch List.
- **Import** button to import a list of PANs to add to the list
- **Add** button to add an individual PAN to the list

> **Note**
>
> The supported file formats for uploading rule configurations are .csv and .xml.

To search for a cardholder PAN:

- Enter a **Primary Account Number** and click the **Search** button. Leave the field blank to search for all PANs.

The **Soft Launch List** page is displayed.

Soft Launch List

Rules > Rule Management > Authentication > Soft Launch List > Soft Launch List Search Results

PANs are listed according to the search criteria you entered on the **Search Soft Launch List** page. You can add an individual PAN or import a list of PANs to the list or select any PAN to delete or edit it.

The following fields and links are displayed:

- **Select** checkbox - for selecting the Primary Account Number in conjunction with the **Delete** button
- **Primary Account Number** link to the **Edit Soft Launch List** page
- Links for **Add** and **Import**

To edit a PAN:

- Click the **Primary Account Number** link.

  The **Edit Soft Launch List** page is displayed.

To add an individual PAN:

- Click the **Add** link.

  The **Add Soft Launch List** page is displayed.

To import a file of PANs:

- Click the **Import** link.

  The **Import Soft Launch List** page is displayed.

To delete a PAN:

- Select the checkbox adjacent to the Primary Account Number and click the **Delete** button.

Edit Soft Launch List

Rules > Rule Management > Authentication > Soft Launch List > Search Soft Launch List > Edit Soft Launch List

To edit the Soft Launch List:

- **Issuer** name is displayed and cannot be changed.
- Edit the **Primary Account Number**.
- Click the **Apply** button.

  A confirmation message will be displayed.
- Click the **Back** button to return to the Soft Launch List Search Results.

Add to Soft Launch List

Rules > Rule Management > Authentication > Soft Launch List > Add Soft Launch List

This page is used to add individual PANs to the Soft Launch List.

To add to the Soft Launch List:

- **Issuer** name is displayed and cannot be changed.
- Enter a **Primary Account Number**.
- Click the **Apply** button.

  A confirmation message will be displayed.

Import Soft Launch List

Rules > Rule Management > Authentication > Soft Launch List > Import Soft Launch List

This page is used to import a file of cardholder PANs to add to the Soft Launch List. The format of the file should be CSV, with each record incorporating a PAN value.

To import a file of PANs:

- **Issuer** name is displayed and cannot be changed.
- Click the **Choose File** / **Browse...** button adjacent to **File name**, to locate and select a file to import.

  The **No file chosen** message will be replaced with the name of the file to be imported.

> **Note**
>
> A maximum of 1000 records can be imported at one time.

- Click the **Apply** button.

  A confirmation message will be displayed.

Merchant Whitelist Rule

If any of the Merchant details are on the Merchant Whitelist, the authentication procedure will be bypassed and the other business rules will be applied in the specified sequence.

> **Note**
>
> If any of the merchant's details are on the list, the cardholder will be considered to be transparently authenticated and transaction status will set to **Y** or **A**, as configured under the **Settings** tab.

When checking the Merchant details from a transaction against the Merchant Whitelist, the Merchant is considered to be a match if any of the Merchant details (Merchant ID, Merchant name, Acquirer BIN or Merchant country) from the PAReq match the corresponding details on the Whitelist.

You can search for Merchant details in the Merchant Whitelist, import a file of merchant details or add Merchant details to the list and edit Merchant details.

The first page in this section is **Search Merchant Whitelist**.

Search Merchant Whitelist

Rules > Rule Management > Merchant Whitelist > Search Merchant Whitelist

This page displays:

- **Search** button to search for Merchant details already in the Merchant Whitelist.
- **Import** button to import a file of Merchant details to add to the list

> **Note**
>
> The supported file formats for uploading rule configurations are .csv and .xml. The following is an example of a sample .xml file:

```
<MerchantWhitelist>
      < Merchant>
          <Id>1</Id>
```

```
            <Name>hi</Name>
            <AcquirerBIN>412345</AcquirerBIN>
            <CountryCode>024</CountryCode>
        </Merchant>
        <Merchant>
            <Id>12134567890</Id>
            <Name>hi</Name>
            <AcquirerBIN>412345</AcquirerBIN>
        </Merchant>
    </MerchantWhitelist>
```

- **Add** button to add individual Merchant details to the list

To search for a Merchant:

- **Issuer** name is displayed and cannot be changed.
- Enter Merchant details as follows, or leave them blank to display all Merchants on the Whitelist:
  - Enter a **Merchant ID**
  - Enter a **Merchant name**
  - Enter an **Acquirer BIN**
  - Select a **Merchant country** from the drop down list. Default is All.
- Click the **Search** button.

The **Merchant Whitelist** page is displayed.

Merchant Whitelist

Rules > Rule Management > Authentication > Merchant Whitelist > Merchant Whitelist Search Results

Merchant details are listed according to the search criteria you entered on the **Search Merchant Whitelist** page. You can add an individual Merchant or import a file of Merchants to the list or select any Merchant to delete or edit it.

The following fields and links are displayed:

- **Select** checkbox for selecting one or more sets of Merchant details
- **Delete** button used in conjunction with selected Merchant details

- *Merchant Id*, *Merchant Name*, *Acquirer BIN* and *Merchant Country* links to the **Edit Merchant Whitelist** page
- Links for *Add* and *Import*

To add individual Merchant details:

- Click the *Add* link.

  The **Add Merchant Whitelist** page is displayed.

To import a file of Merchant details:

- Click the *Import* link.

  The **Import Merchant Whitelist** page is displayed.

To delete Merchant details:

- Select the checkbox adjacent to the Merchant ID and click the **Delete** button.

  Confirmation of the deletion is displayed.

Edit Merchant Whitelist

Rules > Rule Management > Authentication > Merchant Whitelist > Edit Merchant Whitelist

To edit the Merchant Whitelist:

- **Issuer** name is displayed and cannot be changed.
- Edit any of the Merchant fields:
  - **Merchant ID**
  - **Merchant Name**
  - **Acquirer BIN**
  - **Merchant Country**.
- Click the **Apply** button.

  A confirmation message will be displayed.
- Click the **Back** button to return to the Merchant Whitelist Search Results.

Add to Merchant Whitelist

Rules > Rule Management > Authentication > Merchant Whitelist > Add to Merchant Whitelist

This page is used to add Merchant details to the Merchant Whitelist.

When checking whether a transaction matches the Merchant Whitelist, the system will compare the Merchant Id, Merchant Name, Acquirer BIN and Merchant Country from the PAReq, with the values entered.

To add to the Merchant Whitelist:

- **Issuer** name is displayed and cannot be changed.
- Enter a value into any of the Merchant fields:
  - **Merchant Id**
  - **Merchant Name**
  - **Acquirer BIN**
  - **Merchant Country** select from the drop down list.
- Click the **Apply** button.

  A confirmation message will be displayed.
- Click the **Back** button to return to the Merchant Whitelist Search Results.

Import Merchant Whitelist

Rules > Rule Management > Authentication > Merchant Whitelist > Import Merchant Whitelist

This page is used to import a file of merchant whitelist rules to add to the Merchant Whitelist. The format of the file should be CSV, with each record incorporating values for any of the required Merchant Id, Merchant Name, Acquirer BIN and Merchant Country fields necessary to define the rule.

Note

Merchant country is defined by a country code and name and uses the format:
< ISO 3166-1 numeric code > / < ISO 3166-1 country name>

For example:

008/Albania

024/Angola

036/Australia

To import a file of merchant whitelist rules:

- **Issuer** name is displayed and cannot be changed.
- Click the **Choose File** / **Browse…** button adjacent to **File name**, to locate and select a file to import.

  The **No file chosen** message will be replaced with the name of the file to be imported.

  > **Note**
  >
  > A maximum of 1000 records can be imported at one time.

- Click the **Apply** button.

A confirmation message will be displayed.

Merchant Watchlist

If any of the Merchant's details are on the Merchant Watchlist, the authentication procedure will be initiated. If the Merchant's details are not on the list, the other business rules will be applied in the specified sequence.

> **Note**
>
> If any of the merchant details are on the list, the authentication procedure will be initiated.

When checking the Merchant details from a transaction against the Merchant Watchlist, the Merchant is considered to be a match if any of the Merchant details (Merchant ID, Merchant name, Acquirer BIN or Merchant country) from the PAReq match the corresponding details on the Watchlist.

You can search for Merchant details in the Merchant Watchlist, import a file of Merchant details or add individual Merchant details to the list and edit Merchant details.

The first page in this section is **Search Merchant Watchlist**.

Search Merchant Watchlist

Rules > Rule Management > Authentication > Merchant Watchlist > Search Merchant Watchlist

This page displays:

- **Search** button to search for Merchant details already in the Merchant Watchlist.
- **Import** button to import a file of Merchant details to add to the list

> **Note**
>
> The supported file formats for uploading rule configurations are .csv and .xml

- **Add** button to add individual Merchant details to the list

To search for a Merchant:

- **Issuer** name is displayed and cannot be changed.
- Enter Merchant details as follows, or leave them blank to display all Merchants on the Watchlist:
  - Enter a **Merchant ID**
  - Enter a **Merchant name**
  - Enter an **Acquirer BIN**
  - Select a **Merchant country** from the drop down list. Default is All.
- Click the **Search** button.

The **Merchant Watchlist** page is displayed.

Merchant Watchlist Search Results

Rules > Rule Management > Authentication > Merchant Watchlist > Merchant Watchlist Search Results

Merchant details are listed according to the search criteria you entered on the **Search Merchant Watchlist** page. You can add an individual Merchant or import a file of Merchants to the list or select any Merchant to delete or edit it.

The following fields and links are displayed:

- **Select** checkbox for selecting one or more sets of Merchant details
- **Delete** button used in conjunction with selected Merchant details

- **Merchant Id**, **Merchant Name**, **Acquirer BIN** and **Merchant Country** links to the **Edit Merchant Watchlist** page
- Links for **Add** and **Import**

To add individual Merchant details:

- Click the **Add** link.

  The **Add Merchant Watchlist** page is displayed.

To import a file of Merchant details:

- Click the **Import** link.

  The **Import Merchant Watchlist** page is displayed.

> **Note**
>
> The supported file formats for uploading rule configurations are .csv and .xml

To delete Merchant details:

- Select the checkbox adjacent to the Merchant ID and click the **Delete** button.

  Confirmation of the deletion is displayed.

Edit Merchant Watchlist

Rules > Rule Management > Authentication > Merchant Watchlist > Edit Merchant Watchlist

To edit the Merchant Watchlist:

- **Issuer** name is displayed and cannot be changed.
- Edit any of the Merchant fields:
  - **Merchant Id**
  - **Merchant Name**
  - **Acquirer BIN**
  - **Merchant Country**.
- Click the **Apply** button.

  A confirmation message will be displayed.

- Click the **Back** button to return to the Merchant Watchlist Search Results.

Add to Merchant Watchlist

Rules > Rule Management > Authentication > Merchant Watchlist > Add to Merchant Watchlist

This page is used to add Merchant details to the Merchant Watchlist.

A new Merchant Watchlist rule can be created by entering a value in at least one of the fields on the page:

When checking whether a transaction matches the Merchant Watchlist, the system will compare the Merchant Id, Merchant Name, Acquirer BIN and Merchant Country from the PAReq, with the values entered.

To add to the Merchant Watchlist:

- **Issuer** name is displayed and cannot be changed.
- Enter a value into any of the Merchant fields:
  - **Merchant ID**
  - **Merchant Name**
  - **Acquirer BIN**
  - **Merchant Country** select from the drop down list.
- Click the **Apply** button.

  A confirmation message will be displayed.
- Click the **Back** button to return to the Merchant Watchlist Search Results.

Import Merchant Watchlist

Rules > Rule Management > Authentication > Merchant Watchlist > Import Merchant Watchlist

This page is used to import a file of merchant watchlist rules to add to the Merchant Watchlist. The format of the file should be CSV, with each record incorporating values for any of the required Merchant Id, Merchant Name, Acquirer BIN and Merchant Country fields necessary to define the rule.

> **Note**
>
> Merchant country is defined by a country code and name and uses the format:
>
> < ISO 3166-1 numeric code> < ISO 3166-1 country name>
>
> For example:
>
> 008/Albania
>
> 024/Angola
>
> 036/Australia

To import a file of merchant watchlist rules:

- **Issuer** name is displayed and cannot be changed.
- Click the **Choose File** / **Browse** button to locate and select the **File name**.

  The **No file chosen** message will be replaced with the name of the file to be imported.

  > **Note**
  >
  > A maximum of 1000 records can be imported at one time.

- Click the **Apply** button.

  A confirmation message will be displayed.

Location Watchlist

If the merchant's country is on the Location Watchlist, the authentication procedure will be initiated. If the merchant's country is not on the list, the remaining business rules will be applied.

> **Note**
>
> If the merchant's country is not on the list, the cardholder will be considered to be transparently authenticated and transaction status will set to **Y** or **A**, as configured under the **Settings** tab.

You can search for Locations, import a file of Locations or add an individual Location to the list.

The first page in this section is **Search Location Watchlist**.

Search Location Watchlist

Rules > Rule Management > Authentication > Location Watchlist > Search Location Watchlist

This page displays:

- **Search** button to search for countries already in the Location Watchlist.
- **Import** button to import a file of countries to add to the list

> **Note**
>
> The supported file formats for uploading rule configurations are .csv and .xml.

- **Add** button to add individual countries to the list

To search for a Location:

- **Issuer** name is displayed and cannot be changed.
- Select a **Merchant country** from the drop down list. Default is All.
- Click the **Search** button.

  The **Location Watchlist** page is displayed.

Location Watchlist Search Results

Rules > Rule Management > Authentication > Location Watchlist > Location Watchlist Search Results

Merchant countries are listed according to the search criteria you entered on the **Search Location Watchlist** page. You can add an individual Merchant country or import a file of Merchant countries to the list or select any Merchant country to delete or edit it.

The following fields and links are displayed:

- **Select** checkbox for selecting one or more Merchant Countries
- **Delete** button used in conjunction with selected Merchant Countries
- **Merchant Country** link to the **Edit Location Watchlist** page
- Links for **Add** and **Import**

To add individual Merchant countries:

- Click the **Add** link.

  The **Add Location Watchlist** page is displayed.

To import a file of Merchant countries:

- Click the **Import** link.

  The **Import Location Watchlist** page is displayed.

> **Note**
>
> The supported file formats for uploading rule configurations are .csv and .xml.

To delete Merchant countries:

- Select the checkbox adjacent to the Merchant Country and click the **Delete** button.

  Confirmation of the deletion is displayed.

Edit Location Watchlist

Rules > Rule Management > Authentication > Location Watchlist > Edit Location Watchlist

Editing a Merchant country replaces the selected country with the new country selected.

To edit the Location Watchlist:

- **Issuer** name is displayed and cannot be changed.
- Edit **Merchant country** by selecting a different country from the drop down list.
- Click the **Apply** button.

  A confirmation message will be displayed.
- Click the **Back** button to return to the Location Watchlist Search Results.

Add to Location Watchlist

Rules > Rule Management > Authentication > Location Watchlist > Add to Location Watchlist

This page is used to add individual Merchant countries to the Location Watchlist.

When checking whether a transaction matches the Merchant Watchlist, the system will compare the Merchant Country from the PAReq, with the Merchant countries entered on the watchlist.

To add to the Location Watchlist:

- **Issuer** name is displayed and cannot be changed.
- Select a **Merchant Country** from the drop down list.
- Click the **Apply** button.

  A confirmation message will be displayed.
- Click the **Back** button to return to the Merchant Watchlist Search Results.

Import Location Watchlist

Rules > Rule Management > Authentication > Location Watchlist > Import Location Watchlist

This page is used to import a file of merchant countries to add to the Location Watchlist. The format of the file should be CSV, with each record incorporating a Merchant country value.

> **Note**
>
> Merchant country is defined by a country code and name and uses the format:
> < ISO 3166-1 numeric code>/< ISO 3166-1 country name>
>
> For example:
>
> 008/Albania
>
> 024/Angola
>
> 036/Australia

To import a file of Location watchlist rules:

- **Issuer** name is displayed and cannot be changed.
- Click the **Choose File** / **Browse** button to locate and select the **File name**.

  > **Note**
  >
  > A maximum of 1000 records can be imported at one time.

- Click the **Apply** button.

A confirmation message will be displayed.

Domestic & International Transaction Amount Threshold

Rules > Rule Management > Authentication > Domestic & International Transaction Amount Threshold

The domestic and international transaction thresholds determine if the authentication procedure is to be initiated or bypassed.

The Domestic threshold is used when the transaction purchase currency is the selected Default Currency; otherwise, the International threshold is used.

If the transaction amount is less than the defined transaction amount threshold, the cardholder will be transparently authenticated.

If the transaction amount is greater than or equal to the threshold, the authentication procedure will be initiated.

> **Note**
>
> If the transaction currency is not in the currency file, the default will be for the Issuer ACS to pass the PARes back to the merchant indicating Cardholder Transparently Authenticated.

The following fields and links are displayed:

- **Select** checkbox for selecting one or more sets of rules.
- **BIN** links to the **Edit Domestic & International Transaction Amount Threshold** page.
- **Delete** button used in conjunction with selected rules.
- Links for **Manage Exchange Rates** and **Add**.

To manually create currency exchange values:

- Click the **Manage Exchange Rates** link.

  The **Manage Exchange Rate** page is displayed.

To add Domestic & International Transaction Amount Threshold rules:

Rules > Rule Management > Authentication > Authentication > Domestic & International Transaction Amount Threshold > Add

- Click the **Add** link.

  The **Add Domestic & International Transaction Amount Threshold** page is displayed.

> **Note**
>
> The *Add* link is disabled when a rule has been set for All BINs of the selected issuer.

To delete a rule:

- Select the checkbox adjacent to the BIN and click the **Delete** button.

  Confirmation of the deletion is displayed.

To set Domestic & International transaction amount thresholds:

- **Issuer** name is displayed and cannot be changed.
- Select the **BIN** to be used for the threshold from the drop down list.
- Enter a **Domestic amount** for the threshold.
- Enter an **International amount** for the threshold.
- Select the currency to be used for the domestic threshold from the **Default currency** drop down list.
- Click the **Apply** button to save changes.

  A confirmation message will be displayed.

Manage Exchange Rates

Rules > Rule Management > Authentication > Domestic & International Transaction Amount Threshold > Manage Exchange Rates

Currency exchange values can be manually defined for each issuer to set customised exchange rates or for rates not available on the automated list. These rates take precedence over the general rates that are downloaded from external resources or manually defined by the system administrator in System Management > Exchange Configuration.

The following fields and links are displayed:

- **Select** checkbox for selecting one or more defined exchange rates.

- *Base Currency* links to the **Edit Exchange Rate** page.
- **Delete** button used in conjunction with selected exchange rates.
- **Add** link to add an exchange rate.

To manually create currency exchange values:

Rules > Rule Management > Authentication > Domestic & International Transaction Amount Threshold > Manage Exchange Rates > Add

- Click the *Add* link.

  The **Add Exchange Rate** page is displayed.

To delete a defined exchange rate:

- Select the checkbox adjacent to the Base Currency and click the **Delete** button.

  Confirmation of the deletion is displayed.

To manually create currency exchange values:

- **Issuer** name is displayed and cannot be changed.
- Select the **Base currency** and **Target currency** to be used for the currency exchange rate from the drop down list.
- Enter a **Rate** for the currency exchange.
- Click the **Apply** button to save changes.

  A confirmation message will be displayed.

Stand-In Transaction Threshold

Rules > Authentication Exemption > Stand-In Transaction Threshold

The Stand-In Transaction Threshold rule is applied if the remote authentication server (CAAS) cannot be contacted or is not responding at the Verify Registration (first message to CAAS) stage.

However, if the cardholder is in the middle of the authentication process and has commenced selection and the CAAS cannot be contacted or is not responding then the Issuer ACS will be unable to authenticate the transaction and Transaction Status will be set to 'U' (Unable to authenticate).

> **Note**
>
> If the transaction amount (AUD or converted to AUD) is less than the Stand-In threshold, the Issuer ACS will pass the PARes back to the merchant indicating Cardholder Transparently Authenticated.

The following fields and links are displayed:

- **Select** checkbox for selecting one or more sets of rules.
- *BIN* links to the **Edit Stand-In Transaction Threshold** page.
- **Delete** button used in conjunction with selected rules.
- Links for **Manage Exchange Rates** and **Add**.

To manually create currency exchange values:

- Click the **Manage Exchange Rates** link.

  The **Manage Exchange Rate** page is displayed.

To add Stand-In Transaction Threshold rules:

Rules > Authentication Exemption > Stand-In Transaction Threshold > Add

- Click the **Add** link.

  The **Add Stand-In Transaction Threshold** page is displayed.

  > **Note**
  >
  > The *Add* link is disabled when a rule has been set for All BINs of the selected

  issuer.

To delete a rule:

- Select the checkbox adjacent to the BIN and click the **Delete** button.

  Confirmation of the deletion is displayed.

To set Stand-In Transaction thresholds:

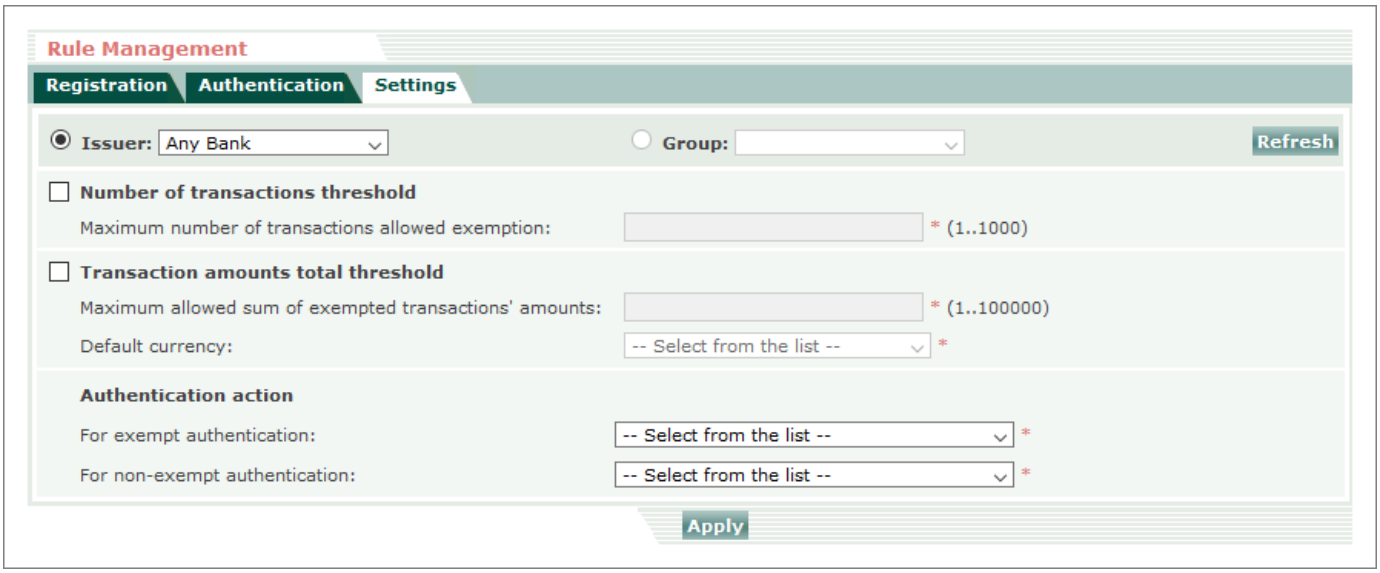
- **Issuer** name is displayed and cannot be changed.
- Select the **BIN** to be used for the threshold from the drop down list.

- Enter an **Amount** for the threshold.
- Select the default currency to be used for the threshold from the **Currency** drop down list.
- Click the **Apply** button to save changes.

  A confirmation message will be displayed.

Settings

Rules > Rule Management > Settings



This section is used to set thresholds, by issuer or issuer group, for the number of transactions and the transaction amounts total (which determine if the authentication procedure is to be initiated or bypassed). It is also used to define the authentication response (PARes) and transaction status for cardholders when authentication is bypassed.

If one threshold is specified - if the **number of transactions or** the **transaction amounts total** is less than the specified threshold, the cardholder will be transparently authenticated.

If both thresholds are specified, if the **number of transactions and** the **transaction amounts total** are less than the specified thresholds, the cardholder will be transparently authenticated.

If the **number of transactions or** the **transaction amounts** total is greater than or equal to the specified thresholds, the authentication procedure will be initiated.

Use the following fields to view / edit the Settings:

- If the required **Issuer** or an **Issuer Group** is not displayed, select it from the appropriate drop down list.
- Click the **Refresh** button.

> **Note**
>
> These fields are not displayed if the user is assigned to a single issuer.

- To set a **Number of transactions threshold**, select the checkbox and enter a number from **1** to **1000** (inclusive) in **Maximum number of transactions allowed exemption**. This is the number of times a cardholder can be exempt, after which time they will be required to be authenticated. Once authenticated successfully, the transaction count is reset to 0.
- To set a **Transaction amounts total threshold**, select the checkbox and enter an amount from **1** to **100000** (inclusive) in **Maximum allowed sum of exempted transactions amounts**. Once the cardholder is authenticated successfully, the transaction amounts total is reset to 0.
- Select the **Default** currency, which will apply to the transaction amounts total, from the drop down list.
- To select an **Authentication action** for exempt and non-exempt authentication, select from the appropriate drop down lists for the selected issuer or issuer group:
- **For exempt authentication** - the options are:

  Set PARes='A' (Attempted)

  Set PARes='Y' (Approved)
- **For non-exempt authentication** - the options are:

  Set PARes='N' (Not approved)

  Show authentication page
- Click the **Apply** button to save changes.

Admin Users

System Administrators and **Member Administrators**

This section is used to set up and manage administrative users. A pre-defined access level group assigned to the username determines the access level of each administrative user. A **read only** option is available for each access level group to provide access to the appropriate sections for support roles that are not required to add records, edit details or upload files.

When the Issuer system is first installed, it creates the main administrative user, named **administrator**, by default. The **administrator** user has the highest level of access throughout the issuer system and can create other users that have restricted access rights, are restricted to certain tasks or are have limited access to a certain issuer.

The access level groups are:

- **System administrator**

  This is the highest level of access in the system with access to system options, issuer management, user management, cardholder management, transactions, reporting and audit log.

- **Issuer administrator**

  This level provides access to issuer configuration options, cardholder management, transactions, reporting and audit log.

- **IT security administrator**

  This level provides dedicated access to the **Audit Log**, for an issuer or issuer group

- **Member administrator**

  This level provides dedicated access to the **Admins** section (administration user management), for an issuer or issuer group

- **Business administrator**

The business level of access to the system provides access to cardholder management, transactions, reporting and audit log.

- **Helpdesk**

  The helpdesk user can access cardholder management and transactions.

> **Note**
>
> If the issuer has access to business rules functionality, **Business admin** and / or **Helpdesk** users can be granted access to the Rules section, refer to Section 3.1.3.4 - Issuer Details for further information.

The ActiveAccess issuer system is designed for simultaneous use by multiple issuers, with each access level being able to be restricted to a certain issuer or a group of issuers. This provides the flexibility of allowing a third party to manage multiple issuers on their behalf as well as allowing each issuer to manage their own system without having access or interfering with other issuers.

Normally when a new issuer signs up with the system, the system administrator creates a new issuer and a new issuer administrator. The issuer administrator can then create business administrator and helpdesk users as appropriate for their requirements.

***Admins** has the following menu options:*

- **Find Admin** for maintaining administrative users and their details
- **New Admin** for adding new administrative users

Find Admin

This page allows you to search for an administrative user based on Status, Group, Username, Full name, Issuer or Issuer group.

Admins > Find Admin

Use the following fields to search for an Admin user:
You can leave all fields at default or blank to display a list of all admin users.

- Select a **Status** or **All** from the drop down list. You can search for enabled or disabled administrative users or both.

- Select a **Group** or **All** from the drop down list. Depending on your access level you may be able to search for **System Admin**, **Issuer Admin**, **IT Security, Member Admin**, **Business Admin** or **Helpdesk** users.
- Enter all or part of the administrator's **Username**.
- Enter all or part of the administrator's **Full name.**
- Select an **Issuer** or **All** from the drop down list.
- Select an **Issuer Group** or **All** from the drop down list.
- ***Search*** button to display results

The **Search Result** page will be displayed.

Admin Search Results

Admins > Find Admin > Search Results

Administrative users are listed according to the search criteria you entered on the **Find Admin** page. You can select any administrative user and delete, enable or disable them.

> **Note**
>
> The main system administrator (administrator) is not displayed in the search results and cannot be disabled or removed.

You can also browse to the admin details page by following the link under **Username** or **Full name** and you can also use the **Change password** link to reset a user's password.

Use the following steps to delete, enable or disable an administrative user:

- Choose one or more users by clicking the **Select** checkbox adjacent to the **Username**
- Click the **Delete**, **Enable** or **Disable** button as appropriate.

  A confirmation message will be displayed.

Use the following steps to select an administrative user:

- Click the **Username** hyperlink for the user for which you wish to view or edit details.

  The **Admin Details** page is displayed.

Use the following steps to change an administrative user's password:

- Click the **Change password** hyperlink for the user whose password you wish to change.

  The **Change Password** page is displayed.

*Use the following steps to select **all** items that match the search criteria:*

- Click the checkbox under the **Select** column to select or unselect all items. This allows you to perform the desired action on all selected items.

> **Warning**
>
> **Important:** The display of search results is limited to 400 records, however if you select all records, all records matching the search criteria will be affected by the action you choose to perform.

> **Warning**
>
> Performing the selected action on a large number of records may take a long time to complete and will generate the equivalent number of audit log records. Use this functionality on a large number of records diligently and only where strictly necessary.

Admin Details

Admins > Find Admin > Search Results > Admin Details

This page allows administrative personnel with the appropriate access rights to update administrative user information.

> **Note**
>
> System administrators have access to all admin users. Issuer administrators have access to business admin and helpdesk users.

The following admin details are displayed on this page:

- **Status**

  Can be either **enabled** or **disabled**. A user in a disabled state will not be able to login to the system.

> **Note**
>
> The main administrator cannot be disabled.

- **Last login**

  Shows the date and time of last login by the user of this administrative account.

- **Group**

  Indicates the level of access a user has in the administration server and cannot be changed. There access levels are:

  - **System Admin**
  - **Issuer Admin**
  - **IT Security Admin**
  - **Member Admin**
  - **Business Admin**
  - **Helpdesk**.

  You can create a user at any one of these access levels with **Read only** access by selecting the Read only checkbox below.

  Only administrators that belong to the System Admin and Issuer Admin groups have access to the Admins section and can create new admin users. Issuer Admin group users may only create Business Admin and Helpdesk users. System Admin users may create users at any and all levels.

> **Note**
>
> The main administrator's group cannot be changed.

- **Issuer** or **Issuer Group** radio button

  Specifies which issuer or issuer group the issuer admin user can access. Issuer admin users may be assigned to a previously defined issuer group rather than a single issuer, which enables them to manage multiple issuers.

  Administrators who belong to System Admin group can always access all issuers and as such, issuer selection for system administrators is not required.

- **Username**

A unique name used to identify the administrative user and used for logging into the administration server. The main administrator's username is always **administrator** and cannot be changed.

- **Full name**

  Optional user information that is stored for housekeeping purposes.

- **Email address**

  Optional user information that is stored for housekeeping purposes.

- **Contact number**

  Optional user information that is stored for housekeeping purposes.

- **Address**

  Optional user information that is stored for housekeeping purposes.

- **Change Password** (***Admins > Find Admin > Search Results > Change Password***)

  While administrators with a higher access level cannot access or see other admin passwords, they can reset or change other users' password. The newly selected password may only be valid for first login if "User must change password at next logon" option is selected.

- **Read only access** checkbox

  Select this checkbox if the user performs a support role that is not required to add records, edit details or upload files, for example.

- **Two-factor authentication login** checkbox

  Select this checkbox if you want to enable two-factor authentication when this user logs in.

> **Note**
>
> An email will be sent to the user with a QR code, to be used with Google Authenticator. To use this option, mail server must be configured in ***System Management > Settings***. For more information, refer to Login.

- Click the **Apply** button to save changes.

  A confirmation message will be displayed.

  OR

  Click the **Back** button to return to the **Search Results** page without saving any changes.

Use the following steps to change an administrative user's password:

- **Username** is displayed and cannot be changed.
- Enter the new **Password.**
- Confirm the new password in the **Re-enter new password** field.
- Click the **User must change the password at next logon** checkbox, if required.
- Click the **Apply** button to save changes.

  A confirmation message will be displayed.

New Admin

Admins > New Admin

All ActiveAccess administrative users must be set up in this section.

Creating a new administrative user:

- **Status**

  Can be either **enabled** or **disabled**. A user in a disabled state will not be able to login to the system.

- **Group**

  Indicates the level of access a user has in the administration server. There access levels are:

  - **System Admin**
  - **Issuer Admin**
  - **IT Security Admin**
  - **Member Admin**
  - **Business Admin**
  - **Helpdesk**

  You can create a user at any one of these access levels with **Read only** access by selecting the **Read only access** checkbox.

  Only administrators that belong to the System Admin and Issuer Admin groups have access to the Admins section and can create new admin users.

System Admin users may create users at any and all levels. There is no interdependency between System Admin users and the other users they create.

- **Issuers** or **Issuer Groups**

  Choose the appropriate radio button and select an **Issuer** or an **Issuer group** from the drop down list.

  Administrators who belong to the System Admin group can always access all issuers and as such, issuer selection for system administrators is not required.

- **Username**

  A unique name used to identify the administrative user and used for logging into the administration server. The main administrator's username is always **administrator** and cannot be changed.

- **Password**

  Enter a password

- **Re-enter password** to confirm it.

- Select the **User must change password at next logon** checkbox if you want this password to be valid for the user's first login only.

- Select the **Two-factor authentication login** checkbox if you want to enable two factor authentication when this user logs in.

> **Note**
>
> An email will be sent to the user with a QR code, to be used with Google Authenticator. To use this option, mail server must be configured in ***System Management > Settings***. For more information, refer to Login.

- **Full name**

  Optional user information that is stored for housekeeping purposes.

- **Email address**

  May be used by the system in order to send email notifications, if the appropriate option is configured by the system administrator.

  If `Two-factor authentication login` is enabled, this email address will be used for sending a QR code to the user. Mail server must be configured in ***System Management > Settings***.

- **Contact number**

Optional user information that is stored for housekeeping purposes.

- **Address**

  Optional user information that is stored for housekeeping purposes.

- **Read only access** checkbox

  Select this checkbox if the user performs a support role that is not required to add records, edit details or upload files, for example.

- Click the **Apply** button to save changes.

  A confirmation message will be displayed.

> **Note**
>
> For further information on individual fields, please refer to Admin Details.

Cards

System Administrators, **Issuer Administrators**, **Business Administrators**, **Helpdesk Users**

This section is used for registering and maintaining individual cards. You can search for cards; enable or disable cards; view card information (including the enrolment status); update card information; and pre-register new cards.

Info

Please see **Upload Registration Files** in the **Issuers** section for uploading card data for bulk registration or pre-registration of cardholders.

Note

This page will not be available for remote issuers.

Cards has the following sub menu options:

- **Find Card** - for maintaining cards and card details
- **New Card** - for adding new cards

The first **Cards** page is **Find Card**.

Find Card

Cards > Find Card

This page allows you to access card related information by searching for cards based on name on card, card number, client ID, authentication method, issuer, BIN, enrolment status, card status, device authentication enabled or disabled, device type, device serial number, card ID and pre-registration or registration date range.

Note

- Finding a card using a card number is only possible if you enter the full card number. There is no partial number search or wild card search available.
- Search results display the first 400 cards only.

Use the following fields to search for cards:

- There are two options when searching for a card: by entering the card number or by entering the cardholder name (exactly as embossed on the card) and selecting the issuer from the drop down list.
- Enter the cardholder's full **Name on Card** and select the **Issuer** of the card from the drop down list to view all matching records. The cardholder name is not case sensitive.
- Enter the full **Card number**. Multiple search results are displayed when a card account has more than one cardholder.
- Enter the **Client ID**.
- Enter the full **Authentication Method** to show cards from only one authentication scheme. J/Secure, ProtectBuy, SafeKey, SecureCode and VbV schemes available.
- Select the Issuer from the drop down list or select the Group from the drop down list.
- Select the **BIN** from the drop down list.
- Select the card's **Enrolment Status** from the drop down list. You can choose to search for **All**, **Pre-registered**, **Registered**, or **Re-activated** cards.
- Select the card's **Status** from the drop down list. You can choose to search for enabled, disabled or locked cards.
- Select to search for card for which **Device authentication** enabled or disabled. This allows you to limit the results for cards that support two-factor authentication over 3-D Secure or those that do not.
- Select the **Device type** that has been registered for the Card from the drop down list. You can choose to search for VASCO, SMS, OOB, Email, Decoupled Authenticator.
- Enter the **Device serial number** (unique device identifier) that has been registered for the Card.
- Enter the card's record identifier (**Card ID**) to locate a specific record. This is used for advanced diagnostics where the record identifier is obtained directly from the database.

- Specify an optional date range to limit search results based on the card's **Pre-registration Date**.
- Specify an optional date range to limit search results based on the card's **Registration Date**.
- Click the ***Search*** button.

The **Search Result** page will be displayed.

Exporting cards

An export function, which allows you to download lists, is available for the following:

- Pre-registered cardholders for an issuer or issuer group, using the Confirmation Method
- Cardholders for an issuer or issuer group using the Confirmation Method.

> **Note**
>
> Although the display is limited to the first 400 cards for the selected issuer or issuer group, the full list will be downloaded when the **Export** link is selected.

Use the following fields to find cards to download a list of pre-registered cards:

- Select an **Issuer** or **Issuer Group**
- Select **Pre-registered** as the **Enrolment Status**.
- Click the ***Search*** button

  The **Search Result** page will be displayed, showing the pre-registration date, in addition to the standard fields.
- Click the ***Export*** button to download a file containing the relevant cardholder data.

> **Note**
>
> - Only Issuers or Issuer Groups that are using the Confirmation Method can download a list of pre-registered cards.
> - Exporting is only available to administrators with System Admin and Issuer Admin access level.

Use the following fields to find cards to download a list of cardholders:

- Select an Issuer or Issuer Group

- Select **Registered** as the **Enrolment Status**.
- Click the ***Search*** button

  The **Search Result** page will be displayed, showing the registration date, in addition to the standard fields.
- Click the ***Export*** button to download a file containing the relevant cardholder data.

> **Note**
>
> - Exporting is only available to administrators with System Admin and Issuer Admin access level.
> - It is only possible to download a list of cardholders for Issuers or Issuer Groups that are using the Confirmation Method. Cardholders can be filtered by confirmation status and confirmation date.

Card Search Result

Cards > Find Card > Search Result

Cards are listed according to the search criteria you entered on the **Find Card** page. You can select any card and delete, enable or disable them.

The search result page shows card number, name on card, expiry date, issuer enrolment status and card status. Expiry date is an optional field and is only displayed if it was provided at registration.

The card's enrolment status can be either **pre-registered** or **registered**.

The **Search Result** page may return multiple results for a single card number depending on whether this is an account with multiple cardholders or not. Card numbers with multiple cardholders can be distinguished based on the cardholder name.

> **Note**
>
> The issuer system uses the combination of card number and cardholder name (name on card) as the key identifier for authentication purposes.
>
> Card numbers with different card names are treated independently and as such each cardholder can have their separate authentication data. This also means that enabling/disabling registration are handled separately. *For example if you wish to completely remove a card from the issuer system, be sure to select and remove all cardholders.*

You can browse to the card details page by following the link under **Card Number** or **Name on Card**.

Use the following steps to delete, enable or disable a card:

- Choose one or more cards by clicking the **Select** checkbox adjacent to the **Card Number**
- Click the appropriate ***button***.

  A confirmation message will be displayed.

Use the following steps to select a card:

- Click the **Card Number** hyperlink for the card you wish to view or edit details.

  The **Card Details** page is displayed.

Use the following steps to select all items that match the search criteria:

- Click the box under the **Select** column to select or unselect all items. This allows you to perform the desired task on all selected items.

  You should note that all items matching the search criteria will be affected. This includes items displayed on other pages and even those omitted due to the large number of results (display of search results is limited to a maximum of 400 records).

> **Note**
>
> **Important:** If you are selecting a large number of records, you should remember that the operation can take a long time to complete and will generate an audit log record per affected item. Use this functionality on large number of records with diligence and where only strictly necessary.

Card Details

Cards > Find Card > Search Result > Card Details

The following card details can be viewed/ edited on this page:

- **Issuer**

  Shows card's issuing bank and cannot be changed.
- **BINs** - Displays a list of BINs assigned to the issuer. This field is for information only. Issuer BIN can be modified by an administrator with System Admin access level through **System Management > Issuers > Issuer Details > BIN Management** page.

- **Card ID** - Unique card number, which cannot be changed.
- **Status** - Can be **Enabled**, **Disabled** or **Locked**. A card is enabled when the cardholder is first enrolled.

  For security reasons, administration staff may temporarily disable a card.

  A card may also be locked by the system if multiple unsuccessful authentication attempts are detected.

  If the cardholder is enrolled and the card is disabled or locked, it cannot be used to make authenticated payments.

  If the cardholder is not enrolled, the enrolment process cannot be completed if the card is disabled.

  Cards that are locked by the system can be unlocked by administration staff or after a time-out period, as specified in the issuer settings. A card cannot be manually locked.
- **BIN status** - Shows the BIN status, which is either **Enabled** or **Disabled**, indicating the availability of the 3-D Secure service for the card. Cards with a **Disabled** BIN cannot be enrolled, registered or authenticated.
- **Registration date** - Displayed if cardholder is enrolled.
- **Enrolment** - Shows the enrolment status, which is either registered or pre-registered, along with the **Pre-registration** and **Registration date**. If the Issuer is using the Confirmation method, the **Confirmation status** and **Confirmation** date will also appear in this section.
- **Authentication Method** - Specifies the card's authentication scheme and cannot be changed. Currently **SafeKey**, **ProtectBuy**, **J/Secure**, **SecureCode** and **Verified by Visa** schemes are supported.
- **Card number** - Full card number, partially masked.

> **Note**
>
> Please note that the card number must comply with the Luhn / mod 10 algorithm.

- **Name on Card** - Cardholder name as specified on the card.
- **Client ID** - The client ID of the card.
- **Expiry date** - Card expiry date (mm/yyyy).

> **Note**
>
> The card expiry date is mandatory for Mastercard in 3DS2.

- **Device authentication** (if enabled) - Drop down list shows the status of two-factor authentication for 3-D Secure as Enabled. Select Disabled to disable device authentication for this card and the drop down list will be removed once you click **Apply**.

  A card, for which device authentication is enabled, can use an authentication device in addition to the conventional 3-D Secure password as a second factor of authentication.
- **Number of ADS cancellations** - Shows the number of times a cardholder has refused to complete activation during shopping by either opting out of ADS or cancelling the transaction.
- **ADS proof of attempts granted** - Shows the number of times a cardholder has been granted proof of authentication attempt without being required to complete the activation during shopping process.

> **Note**
>
> Where Business Rules are being used and a rule has matched, this value is set to the maximum automatically and therefore the value may not be a true representation of the number of ADS proof of authentication attempts that have been granted to this cardholder.

- **Extended cardholder information** - Each card is also associated with one or more authentication or data fields. The issuer determines the format and number of these fields. Extended cardholder information is only displayed if the system administrator enables this option in the **Issuer Management** section.

  For example a card may be accompanied by / associated with:

  - A **PAM** (Personal assurance message or the greeting message as required in VbV, J/Secure, ProtectBuy, SafeKey and SecureCode schemes)
  - An **Internet PIN** (for secure online transactions). Fields such as Internet PIN are always displayed masked.
  - **Question** and **Answer** fields used for challenging cardholder before resetting the password.
  - A card's authentication **Password**

- **Assigned Devices** appears for cards for which device authentication is enabled and links to a Search Results page, which displays authentication devices assigned to this card.
- **Account History** links to Account History details page, which shows actions affecting account status or the devices attached to the account.
- **Show Transactions** links to the Search Results page, which displays all transactions for this card.
- **Whitelisting** links to the cardholder's whitelisted details page, which displays all whitelisted merchants for this card. This link will be displayed if the Issuer has enabled this option in BIN Management.
- **Generate Activation Code** link allows the administrator to generate an activation code for a replacement device. The link is only shown if the card's primary device is marked as lost or damaged. The cardholder requires this activation code before they can complete linking the replacement device with an existing account.

Assigned Devices

Cards > Find Card > Search Result > Card Details > Assigned Devices

Assigned Devices displays all devices attached to the selected card. It enables you to assign a new device to a card, remove an assignment or change the status of assigned devices to **Lost**, **Damaged** or **Temporarily disabled**.

The following fields and links are displayed:

- **Device Management** - links to the **Device Management** page for manually assigning a new or existing device to the card, or deleting an existing device.

The following fields and links are displayed for each assigned device:

- **Select** checkbox - for selecting the device to use in conjunction with the ***Remove Assignment*** button or ***Delete*** button. To remove assignment of all devices, click the select checkbox in the column heading and then click the ***Remove Assignment*** button. To completely delete a previously registered device from the system, click the adjacent **Select** checkbox and then click the ***Delete*** button.

  If you select all devices a Warning dialog is displayed asking you to confirm that you want to remove all records that match the search criteria.
- **Device ID** links to the **Device Details** page
- **Assign date** - Date and time device was assigned

- **Serial Number** - The unique device identifier
- **Device type** - The type/make of the device such as VASCO, Email, etc
- **Status** - **Active/Lost/Damaged/Temporarily disabled/Deactivated device type**

  An **Active** device can be used in device authentication.

  If a device is reported lost, stolen, damaged, or temporarily disabled, it must be flagged accordingly. A **Lost** or **Damaged** device can no longer be used for authentication and the cardholder must be issued with a new device.

  A **Temporarily disabled** device allows administrators to generate a new backup token.

  Deactivated device type indicates that the issuer has disabled support for this device.
- **Mark as** -
  - **Lost** - click to change the status of the selected device to **lost**.
  - **Damaged** - click to change the status of the selected device to **damaged**.
  - **Temporarily disabled** - click to change the status of the selected device to **temporarily disabled**.
- **Generate Backup Token** link is only shown when the card's primary authentication device is marked as temporarily disabled. Providing the cardholder with a backup token allows the cardholder to continue using the service while they wait for the replacement authentication device to arrive.

  The application currently supports two mechanisms for generating backup tokens: a replacement password (the default) and SMS.

  A replacement password is a static password that can be used as the second factor of authentication for a limited time and for a limited number of times. Using a static password will not be as secure as using an authentication device. Administrators should only issue backup tokens if allowed by the issuer's security policy and if in line with the issuer's requirements for identifying a cardholder.

  A more secure alternative is to use SMS as the backup token, if supported by the issuer device settings. This allows the admin to temporarily switch the cardholder's authentication process to SMS authentication. The cardholder will need to provide a mobile number to which the second factor of authentication will be sent via SMS.

  Note that the first SMS batch is not sent immediately. The SMS is sent when the cardholder attempts to perform their next authentication. They may have to wait a few minutes, once they attempt to login next time, for the batch SMS to arrive. Once they receive the first batch

SMS, they will continue to receive replacement batch SMS tokens when they use up all the numbers in their current batch.

DEVICE MANAGEMENT

Cards > Card Details > Assigned Devices > Device Management

Device Management provides the option to manually assign a new or existing device to the card, or delete an existing device. This is useful for call centre assisted registration of cards. The cardholder needs to provide a token generated by their device in order to complete the assignment. You can either find an existing device that is already registered in the system and verify it or specify a new device to assign and activate.

- Click the appropriate tab to select the following options:
  - **Find Device**
  - **Assign Existing Device**
  - **Assign New Device**.

Find Device

Cards > Card Details > Assigned Devices > Device Management > Find Device

- Select an **Issuer** or **All** from the drop down list
- Enter a **Creation date** and time (dd/mm/yyyy HH:MM) or specify a date and time range for the search result by entering dates and times in the **From** and **To** fields. The date and time format is dd/mm/yyyy HH:MM. Leave the time field empty if you do not wish to limit your search for a particular time of day.
- Select a **Device type**, such as VASCO, SMS, Email, etc
- Enter a **Device Serial number** or specify a serial number range for the search result by entering serial numbers in the **Start** and **End** fields.
- Click ***Search***

A list of devices matching the search criteria will be displayed.

- To remove a device, click the ***Select*** radio button, adjacent to the appropriate **Device ID**, and click the ***Delete*** button.

  This will remove the device from the system.

Assign Existing Device

Cards > Card Details > Assigned Devices > Device Management > Assign Existing Device

Use the following fields to find an existing device:

- Select an **Issuer** or **All** from the drop down list
- Enter a **Creation date** and time (dd/mm/yyyy HH:MM) or specify a date and time range for the search result by entering dates and times in the **From** and **To** fields. The date and time format is dd/mm/yyyy HH:MM. Leave the time field empty if you do not wish to limit your search for a particular time of day.
- Select a **Device type**, such as VASCO, SMS, Email, etc
- Enter a **Serial number** or specify a serial number range for the search result by entering serial numbers in the **Start** and **End** fields.
- Click ***Search***

  A list of devices matching the search criteria will be displayed.
- Click the **Select** radio button, adjacent to the appropriate **Device ID**, and click the **Apply** button

  The **Verify Device** page is displayed

Assign New Device

Cards > Card Details > Assigned Devices > Device Management > Assign New Device

Use the following fields to assign a new device:

- Select the **Assign New Device** tab

New SMS Device

- Select **SMS** from the **Device type** drop down list.
- Click the **Apply** button.

  The **New SMS Device** page is displayed.
- Select an **SMS centre** from the drop down list.
- Select the **Country calling code** for the country that the mobile number is registered to.
- Enter the **Mobile number** of the cardholder. Mobile number should be no longer than 20 characters, including the Country code. Allowed characters are 0-9, '(', ')', '-' and space.

> **Note**
>
> If the national trunk prefix of the mobile number has been entered (`0` or `1`), these digits will automatically be removed from the start of the mobile number by ActiveAccess.

- Click the **Apply** button

  A token will be sent to the mobile number and the **Activate Device** page will be displayed.

  Existing devices need to be verified and new devices need to be activated, using the token sent to the cardholder's mobile, which is generated by assigning the device.
- Enter the token received by the cardholder's mobile in **Enter the token sent to your mobile number**.
- Click the **Apply** button.

New Email Device

- Select **Email** from the **Device type** drop down list.
- Click the **Apply** button.

  The **New Email Device** page is displayed.
- Enter the **Email address** of the cardholder.
- Click the **Apply** button

  A token will be sent to the email address and the **Activate Device** page will be displayed.

  Existing email addresses need to be verified and new email addresses need to be activated, using the token sent to the cardholder's email address, which is generated by assigning the email address.
- Enter the token received by the cardholder's email address in **Enter the token sent to you by email**.
- Click the **Apply** button.

GENERATE BACKUP TOKEN

Cards > Card Details > Assigned Devices > Generate Backup Token

When the cardholder's primary authentication device is marked as lost or damaged, you can link from the **Card Details > Assigned Devices** page to provide the cardholder with a backup token, which all services can use until a replacement device is received.

- Select the Backup device type:
  - **Default** - supplies a static password that can be used as the second factor of authentication for a limited time and for a limited number of times.
  - **SMS** - displayed only if supported by the issuer device settings. This allows the admin user to temporarily switch the cardholder's authentication process to SMS authentication. The cardholder will need to provide the country calling code and a mobile number to which the second factor of authentication will be sent via SMS. Mobile number should be no longer than 20 characters, including the Country Code. Allowed characters are 0-9, '(', ')', '-' and space.
  - **Email** - displayed only if supported by the issuer's device settings. This allows the admin user to temporarily switch the cardholder's authentication process to Email authentication. The cardholder will need to provide the email address to which the second factor of authentication will be sent via email.
- Click the ***Generate*** button.

  A confirmation message will be displayed.

Show Transactions

Cards > Find Card > Search Result > Card Details > Show Transactions > Show Recent Transactions

Show Transactions allows access to lists of all recent and archived 3-D Secure (payment authentication) transactions matching the selected card.

The following fields and links are displayed:

Show Archived Transactions - links to the **Archived Transactions** page which displays the archived 3-D Secure transactions matching the selected card.

Use the following steps to select a transaction and view its details:

- Click the **Date** (and time) hyperlink for the transaction you wish to view.
- The **Transaction Details** page displays the following fields and links:_

- **Show Recent Transactions** - links to the **Recent Transactions** page which displays the recent 3-D Secure transactions matching the selected card.

Use the following steps to select a transaction and view its details:

- Click the **Date** (and time) hyperlink for the transaction you wish to view.

  The **Transaction Details** page is displayed.

Whitelisting

Cards > Find Card > Search Result > Card Details > Whitelisting

Whitelisting allows access to a list of merchants that have been whitelisted by the cardholder.

The following merchant details are displayed:

- Choose one or more merchants by clicking the **Select** checkbox
- **Acquirer Merchant ID**
- **Merchant Name**
- **MCC** - Merchant Category Code
- **Merchant Country Code**
- Click the ***Remove*** button to remove the selected merchants from the whitelist.

New Card

Cards > New Card

You can use the New Card function to manually register cardholders. This function pre-registers cardholders. Cardholders have to finalise their registration by going through the issuer's standard enrolment process.

Creating a new card:

- Select an **Issuer** from the drop down list of available issuers.

  All issuers that your username is assigned to will be listed here.
- **BINs**

Displays a list of BINs assigned to the issuer. This field is for information purposes. When you enter a new card, the card's BIN number must be one of the existing BINs for the selected issuer.

- Select the **Authentication method** supported by the card
- Currently JCB, Discover, American Express, Mastercard and Visa card schemes are supported.

> **Note**
>
> Please note that selecting the authentication method alone does not guarantee that the card can be used in the specified authentication scheme. Other pre-arrangements may also be required. For example in Verified by Visa, a card may not be able to participate before a valid card range that entails the card has been sent to the directory service.

- Select the **Status** of **Enabled** or **Disabled** from the drop down list.

  A card is normally enabled when the cardholder is first enrolled. The administration staff for security reasons may temporarily disable a card. A card may also be automatically locked by the system itself if multiple unsuccessful authentication attempts are detected.

  When a card is disabled or locked, it cannot be used to make authenticated payments if cardholder is enrolled or alternatively if cardholder has not enrolled yet, the enrolment process cannot be completed before this situation is resolved.
- Enter the full **Card number**

> **Note**
>
> The card number must comply with the Luhn / mod 10 algorithm.

- Enter the cardholder name as specified on the card as **Name on Card**
- Enter the card **Expiry date** using mm/yyyy format

> **Note**
>
> The card expiry date is mandatory for Mastercard in 3DS2.

- Enter information required for any **Extended cardholder information**

- Each card is also associated with one or more authentication or data fields. The issuer determines the format and number of these fields. Extended cardholder information is only displayed if the system administrator enables this option in the Issuer Management section.

  For example, a card may be accompanied by a **PAM** (Personal assurance message or the greeting message as required in VbV, J/Secure, ProtectBuy, SafeKey and SecureCode schemes) or may be associated with a **PIN** (for secure online transactions), etc. Fields such as Internet PIN are always displayed masked.

Transactions

> *The ability to search by 3DS version added.*

System Administrators, Issuer Administrators, Business Administrators, Helpdesk Users

This section is used for accessing 3-D Secure transactions, when required for user support purposes, dispute resolution etc. It has the following menu options:

- **Find 3-D Secure** - to search for and view transactions

Info

It is important to note that in the context of this document, a transaction is an authentication record rather than a financial record. The relationship between the authentication record and the actual authorization record depends on the underlying authentication scheme.

Find 3-D Secure

Transactions > Find 3-D Secure

This page allows you to search for and access 3-D Secure authentication records including the proof of authentication, which can be used for dispute resolution for example.

You can search based on Issuer, Authentication method, Date, Amount, Currency, Account number, Merchant name, Transaction ID and AAV/CAVV and more.

Search fields

The search fields available on this page change based on the **3-D Secure version**(s) selected.

Common Search Fields

The following search fields are common between all 3-D Secure versions.

Use the following fields to search for 3-D Secure Transactions:

- Select a **Target database** from the list to search for **Current** or **Archived** transactions, if archiving has been configured on the system.
- Select at least one **3-D Secure version**, by selecting/deselecting the appropriate checkboxes.

> **Note**
>
> The search fields available on this page change based on the 3-D Secure version(s) selected.

- Specify an **ACS Session ID** to search for a specific transaction.
- Select an **Issuer** (from the list of available issuers) to narrow down the search criteria or you can select **All**. If the Issuer has access to Rules, additional search options will be displayed below.
- Select the **Authentication method** from the drop down list. The options are **J/Secure**, **ProtectBuy**, **SafeKey**, **SecureCode**, **SPA** or **VbV**.
- You can specify an exact **Date** and time or a date and time range (inclusive) in the **From** and **To Date** fields. The date and time format is dd/mm/yyyy HH:MM. Leave the time field empty if you do not wish to limit your search for a particular time of day.

  By default the search is limited to the last 7 days, modify the **From** field if you wish to extend the search period.
- You can specify an exact amount or an amount range (inclusive) in the **From** and **To Amount** fields.
- Select the **Currency** from the drop down list.
- Enter the **Merchant name** in full or in part.
- Enter the full **Merchant ID**.
- Enter the full cardholder **Account number**.
- Enter the **Client ID**.
- Enter the **AAV/CAVV/AEVV** to find the transaction for which this value was generated.

  AAV (Accountholder Authentication Value) / CAVV (Cardholder Authentication Verification Value) / AEVV (American Express Verification Value) is provided in 3-D Secure PARes for J/Secure, ProtectBuy, SafeKey, SecureCode and VbV transactions. AV (Authentication Value) is provided in 3-D Secure 2 ARes and RReq.

- Enter the **Device serial number** to search for transactions that were authenticated using a particular two-factor authentication device.
- Select a **Device type** from the list to search for transactions performed with particular type of authentication device such as SMS or OOB, etc.
- Enter an **Error code/IReq code** to search for in the authentication response message.
- Enter an **Error message/IReq message** to search for in the authentication response message.
- ***Rules*** - displayed if specified Issuer has access to Rules. Where an issuer with **Rules** enabled is selected, additional fields will be available so that you can search for transactions by whether they match or do not match one or more rules.
    - Select one of the following radio buttons:
        - **All** - all transactions, regardless of rules
        - **Matched** - transactions that match the rules selected from the adjacent list
        - **Mismatched** - transactions that do not match the rules selected from the adjacent list
    - Select one or more **Rules** (Ctrl + click to select multiple rules) from the list:
        - **Amount Threshold**
        - **Domestic & International Transaction Threshold**
        - **Location Watchlist**
        - **Merchant Blacklist**
        - **Merchant Watchlist**
        - **Merchant Whitelist**
        - **Soft Launch List**
        - **Stand-In Transaction Threshold**
    - If you have selected the **Matched** radio button, you can enter an **Error code** or **Error message** to search on.
    - If you have selected merchant or location list rules, you can enter additional search parameters:
        - **Merchant ID**
        - **Merchant name**
        - **Acquirer BIN**

- **Merchant country code**

- Click **Search** to display the **Transaction Search Results**.

Additional 3-D Secure version 1 Search Fields

- Enter the **Transaction ID (XID)** as specified by the merchant. Transaction ID can be entered in clear or base64 format but you need to the specify the entire Transaction ID (20 character in clear or 28 characters in base64)
- Select a **Transaction type** from the list to limit the results to transactions that involved a second factor of authentication (**Device over 3-D Secure 1**) or conventional **3-D Secure 1** transactions.
- Select a **VERes status** from the list to search for transactions with a particular verify enrolment result. The **VERes status** can be:
  - **Y** - Cardholder is enrolled
  - **N** - Cardholder is not enrolled
  - **U** - Cardholder enrolment cannot be determined due to technical or other problems
  - **Error** - An error occurred whilst verifying the enrolment status of the cardholder.
- Select a **PARes status** from the list to search for transactions with a particular payer authentication result. The **PARes status** can be:
  - **Y** - Cardholder authentication successful
  - **A** - Cardholder is not enrolled but proof of authentication attempt provided to the merchant
  - **N** - Cardholder authentication failed
  - **U** - Cardholder authentication cannot be completed due to technical or other problems.
  - **N/A** - Cardholder authentication did not complete.
  - **Error** - An error occurred during cardholder authentication.

> **Note**
>
> Use PARes and VERes filters only if you are searching for 3-D Secure records, version 1.0.1 and above.

- Select the transaction **Status** from the list. The transactions can either be **In progress** or **Processed**. An **In progress** status indicates that either the cardholder has not yet finished the authentication process or the system has not yet sent the PATransReq message to the

authentication history server. When you choose to search for all transactions regardless of their status, the system will first return all **In progress** transactions followed by **Processed** transactions. Please note that only the first 400 records are returned in total.

- The **Registered after previous opt-outs or cancellations** option provides a way of searching for and listing transactions where cardholders have completed their registration after initially opting out or cancelling activation during shopping.

Additional 3-D Secure version 2 Search Fields

- Enter an **SDK transaction ID**
- Enter a **DS transaction ID**
- Enter an **3DS Server transaction ID**
- Select an **ARes status** from the list to search for transactions with a particular authentication response. The **ARes status** can be:
  - **Y** - **Authentication / account verification successful**
  - **N** - **Not Authenticated / account not verified** - transaction denied
  - **U** - **Authentication / account verification could not be performed** - technical or other problem, as indicated in ARes or RReq
  - **A** - **Attempts processing performed** - not authenticated / verified but a proof of attempted authentication/verification is provided
  - **C** - **Challenge required** - additional authentication is required using the CReq/CRes
  - **R** - **Authentication / account verification rejected** - Issuer is rejecting authentication / verification and requests that authorisation not be attempted.
- Select a **CRes status** from the list to search for transactions with a particular challenge response. The **CRes status** can be:
  - **Y** - **Authentication / account verification successful**
  - **N** - **Not Authenticated / account not verified** - transaction denied
- Select an **RReq status** from the list to search for transactions with a particular results request. The **RReq status** can be:
  - **Y** - **Authentication / account verification successful**
  - **N** - **Not Authenticated / account not verified** - transaction denied
  - **U** - **Authentication / account verification could not be performed** - technical or other problem, as indicated in ARes or RReq

- **A** - **Attempts processing performed** - not authenticated / verified but a proof of attempted authentication/verification is provided
- **C** - **Challenge required** - additional authentication is required using the CReq/CRes
- **R** - **Authentication / account verification rejected** - Issuer is rejecting authentication / verification and requests that authorisation not be attempted.

- Select a **Message cateory** from the list to search for **All**, **Payment authentication** or **Non payment authentication** transactions.
- Select an **Authentication type** from the list to search for **All**, **Static**, **Dynamic** or **OOB** transactions.
- Select a **Device channel** from the list to search for **App based**, **Browser** or **3DS Requester Initiated** (3RI) transactions.

Transaction Search Results

Transactions > Find Transaction > Search Result

The search result page lists transactions matching the criteria you entered on the **Find Transaction** page and shows transaction date, amount, currency, account number, merchant name, issuer, method, status and transaction type.

Only the first six and the last four digits of the account number are shown. An **X** masks the rest of the digits. You can choose to display card number in plain text in Settings.

You can browse to the transaction details page by following the link under **Account Number**.

Use the following steps to select a transaction and view its details:

- Click the **Date** (and time) hyperlink for the transaction you wish to view.

  The **Transaction Details** page is displayed.

Transaction Details

Transactions > Find Transaction > Search Result > Transaction Details

This page shows the details for the Transaction selected on the Transaction Search Result page.

The following fields can be viewed on this page:

Common Fields

- **Issuer** - Shows the card's issuing bank.
- **Authentication method** - Shows the authentication method relevant to this transaction. The options are: J/Secure, ProtectBuy, SafeKey. SecureCode, or Verified by Visa.
- **Date** - Shows the date and time of the transaction.
- **Amount** - Shows the transaction amount, including the currency.
- **Account number** - Shows the last five digits of the account number, which is used in this transaction. Links to the **Card Details** page.
- **Client ID** - An integer type with 15 digits length assigned to cards that belong to one cardholder.
- **Merchant name** - Shows the merchant name.
- **Merchant ID** - Acquirer-defined merchant identifier, up to 24 characters including the Card Acceptor ID and Card Acceptor Terminal ID.
- **AAV/CAVV/AEVV** - Accountholder Authentication Value / Cardholder Authentication Verification Value / American Express Verification Value for J/Secure, ProtectBuy, SafeKey, SecureCode and VbV authentication. AV (Authentication Value) for 3-D Secure 2.
- **Device serial number** - The unique identifier of the authentication device used in the transaction for two-factor authentication, if available.
- **Device type** - The type of authentication device used in the transaction for two-factor authentication, if available.
- **Transaction type** - 3-D Secure 1, Device over 3-D Secure 1, or 3-D Secure 2, depending on whether the transaction was a conventional password-based 3-D Secure 1 authentication, a two-factor authentication or a 3-D Secure 2 authentication.
- **Error code** - 0 if the authentication request was successfully completed. Any other value indicates an error condition.
- **Error text** - A descriptive message for the response code.
- **Error detail** - Detailed information for the error condition.
- **Matched rules** - Displays the rules against which the transaction matched, with links to the details of the rule at the time of the transaction.

  Where a transaction has matched the merchant blacklist rule, the Matched Rule Details display the matched rule highlighted in red.

- Click on the **Requests and Responses** link to see the details of 3-D Secure 1 messages (VEReq, VERes, PAReq, PARes, PATransReq, PATransRes) or 3-D Secure 2 messages (AReq, ARes, CReq, CRes, RReq, RRes).

Additional 3-D Secure 1 Fields

- **Transaction ID (XID)** - Transaction ID as specified by the merchant (XID for J/Secure, ProtectBuy, SafeKey, SecureCode and VbV)

Additional 3-D Secure 2 Fields

- **SDK transaction ID**
- **DS transaction ID**
- **3DS Server transaction ID**
- **ARes status** - Authentication Result status.
- **ARes status reason** - Authentication Result reason.
- **CRes status** - Challenge Result status.
- **RReq status** - Results Request status.
- **Risk decision** - Action decided by Risk Assessment. Possible values are `Frictionless`, `Frictionless by review (Visa only)`, `Use static password`, `Use Device`, `Use OOB`, and `Rejected`.
- **Failed reason** - Shows the reason for the transaction ending with a status of N or U.
- **Device channel** - Shows the device channel as App based, Browser or 3DS Requester Initiated.
- **Message category** - Shows the message category as Payment authentication or Non payment authentication.
- **Authentication type** - Shows the authentication type as Static, Dynamic or OOB.
- **IAV generation algorithm** - Shows the algorithm used to generate IAV (3DS2 only).

Reports

System Administrators, **Issuer Administrators**, **Business Administrators**

This section provides various reports for authentications, devices, purchase volume, admins, card, enrolment and merchant activity.

Reports can be run for all issuers or any number of issuers or issuer groups. All reports except for the Summary return the total for all selected issuers. The Summary report returns the total and also breaks the report down based on the selected issuers.

Warning

A default time zone is set when the application is installed. This Time zone is displayed, for reference, on the menu bar, from where it can be modified at any time, as and when appropriate. The modification of the Time zone on the menu bar does not change the Time zone for the Issuer.

Note

If you modify the Time zone in the menu bar it will persist for the current session only. It will revert to the Time zone entered in the Issuer settings, the next time you login.

All search parameters for transactions, audit logs and reports (daily, monthly and annual) will be based on the Time zone specified on the menu bar at the time of the search.

IMPORTANT: If the time zone is changed in **Issuers > Settings**, it will impact the data displayed for issuer reports (daily, monthly and annual). When attempting to change the time zone, a warning message is displayed with the following options:

- **Continue and delete report** data - reports will not be available for the selected issuer until the next overnight report run, which will use the new time zone.

> **Note**
>
> If auto archive is enabled, archived data will no longer be collected and previous report data will be lost.

- **Continue and keep report data** - existing report data will be inaccurate due to the time change. Accurate reports will not be available until the next overnight report run, which will use the new time zone.
- **Cancel** - time zone will not be changed.

Reports *section has the following sub menu options:*

- **Card Summary** based reports: authentication attempts, successful authentications, number of enrolled, registered and existing cards and card activity broken down by 3-D Secure provider and selected issuers and issuer groups.
- **Device Summary** based reports: number of device authentication broken down device type for selected issuers and issuer groups.
- **Card Activity** based reports - remaining cards, active cards, and authentication method for selected issuers/issuer groups and time period.
- **Authentication** based reports - Statistics on authentications broken down by the status of associated authentication messages for a given issuer and time period.
- **Enrolment Activity** based reports - enrolled cards, pre-registered cards, cancelled cards and authentication method for a given period.
- **Merchant Activity** based reports - total authentications, authentication status, and authentication method per merchant.
- **Purchases** based reports - purchase volumes, authentication method, and currency type for a given period.
- **Admin** based reports - number of administrators, broken down by user access type and selected issuers and issuer groups.

The first **Reports** page is **Card Summary**.

Card Summary

Reports > Card Summary

The summary report provides an overview of some the more important metrics of the system, including: the number of authentications, successful authentications, number of enrolled, registered and existing cards in the system and card activity. The report is broken down by 3-D Secure provider and can be customised to include one or more issuers or issuer groups. The report can be generated for a specified period of time.

Use the following fields to produce a card summary report:

- Enter a date range in the **From** and **To** dates (dd/mm/yyyy). Defaults are: **From: 01/01/** and **To: 31/12/**.
- Select at least one **Authentication method**, by selecting/deselecting the appropriate checkboxes. All methods are selected by default with the report displaying values against only the selected methods.
- Select which issuers to run the report for; **All Issuers** is selected by default. To run the report for one or more Issuer Groups and/or Issuers, deselect the **All Issuers** checkbox and use the **Add >>**, **<<Remove** buttons to select **Issuers** or **Issuer Groups.**

> **Warning**
>
> Run the report for **All Issuers** with caution as it may take a significant time to produce the report.

- **Extend report by device type** by selecting the checkbox. If selected, at least one device should be selected to run the report.
- **3-D Secure Version** by selecting the **3DS1** and / or **3DS2** checkboxes, as appropriate.
- Click the ***Go*** button to display the new report.
- Click the ***Export*** button in order to export the currently displayed report as a Comma Separated Value (CSV) file.
- Click the ***Back*** button to modify the search criteria.

The following are displayed for the Period specified:

- Date range of report
- Names of the issuer groups / issuers selected or 'for all issuers' if the **All Issuers** checkbox was selected
- Issuer
- Total for Issuers selected

Authentication

- Number of VbV authentication attempts
- Number of SC authentication attempts
- Number of J/S authentication attempts
- Number of SK authentication attempts
- Number of DC authentication attempts
- ***Total number of authentication attempts***
- Number of Successful VbV authentications
- Number of Successful SC authentications
- Number of Successful J/S authentications
- Number of Successful SK authentications
- Number of Successful DC authentications
- ***Total number of successful authentications***

> **Note**
>
> If **Extend by device type** selected, the above are displayed for each device loaded in the system, for example:
>
> - Backup Device
> - VASCO
> - SMS
> - Email
> - Decoupled Authenticator
> - OOB

Enrolment

- Number of pre-registered cards
- Number of fully registered cards
- ***Total number of cards enrolled***

Card Activity

- Number of active VBV cards

- Number of active SecureCode cards
- Number of active J/Secure cards
- Number of active SafeKey cards
- Number of Active ProtectBuy cards
- ***Total Number of active cards***

Device Summary

Reports > Device Summary

This report shows the total number of devices on the system.

Use the following fields to produce a device summary report:

- Select which issuers to run the report for. **All Issuers** is selected by default. To run the report for one or more Issuer Groups and/or Issuers, deselect the **All Issuers** checkbox and use the **Add>>**, **<<Remove** buttons to select **Issuers** or **Issuer Groups.**

> **Warning**
>
> Run the report for **All Issuers** with caution as it may take a significant time to produce the report.

- Select at least one **Device**, by selecting/deselecting the appropriate checkboxes.
- Click the ***Go*** button to display the new report.
- Click the ***Export*** button in order to export the currently displayed report as a Comma Separated Value (CSV) file.
- Click the ***Back*** button to modify the search criteria.

Card Activity

Reports > Card Activity

The card activity report shows the total number of enrolled cards and active cards for a given period for all or selected issuers and/or issuer groups.

The report is also broken down by the authentication method. A card is said to be active in a given period if the cardholder has at least performed one successful authentication with the card in the specified period.

The default report shows monthly cardholder activity for the current year for all issuers. You may select any number of issuers or issuer groups; daily, monthly or annual report period; and specify a date range.

Use the following fields to produce a card activity report:

- Select **Monthly**, **Daily** or **Annual Period** from the drop down list.
- Enter a date range in the **From** and **To** dates (dd/mm/yyyy). Defaults are: **From: 01/01/** and **To: 31/12/**.
- Select at least one **Authentication method**, by selecting/deselecting the appropriate checkboxes. All methods are selected by default with the report displaying values against only the selected methods.
- Select which issuers to run the report for. **All Issuers** is selected by default. To run the report for one or more Issuer Groups and/or Issuers, deselect the **All Issuers** checkbox and use the **Add>>**, **<<Remove** buttons to select **Issuers** or **Issuer Groups.**

> **Warning**
>
> Run the report for **All Issuers** with caution as it may take a significant time to produce the report.

- Click the ***Go*** button to display the report.
- Click the ***Export*** button in order to export the currently displayed report as a Comma Separated Value (CSV) file.
- Click the ***Back*** button to modify the search criteria.

Authentication

Reports > Authentication

This report shows authentication statistics for a given period based on the status codes returned by **VERes** and **PARes** messages for **3DS1** and **AuthRes** messages for **3DS2**.

3DS1 statistics

The status code for **VERes** messages can be **Y** for 'enrolled cards', **N** for 'not enrolled cards' or **U** for 'unable to determine the enrolment status of the card due to some technical difficulty'.

The report calculates **ER** (enrolment rate) as: (VERes=Y) / (VERes=Y+N+U)

The status code for **PARes** messages can be **Y** for 'successful authentication', **N** for 'failed authentication', **A** for 'authentication attempt' or **U** for 'unable to authenticate the card due to some technical difficulty'.

The report calculates **AR** (authentication rate) as: (PARes=Y) / (PARes=Y+A+N+U)

3DS2 statistics

The status code for **ARes** and **CRes** messages can be **Y** for 'enrolled cards', **N** for 'not enrolled cards' or **U** for 'unable to determine the enrolment status of the card due to some technical difficulty'.

The report calculates **AR** (authentication rate) as (ARes=Y+CRes=Y) / (ARes=Y+A+N+U+R+C+I+D)

The default report shows monthly authentication statistics for the current year for all issuers. You may select any number of issuers or issuer groups; a different daily, monthly or annual report period; and specify a date range.

Use the following fields to produce an authentication report:

- Select **Monthly**, **Daily** or **Annual Period** from the drop down list.
- Enter a date range in the **From** and **To** dates (mm/yyyy). Defaults are: **From: 01/** and **To: 12/**.
- Select at least one **Authentication method**, by selecting/deselecting the appropriate checkboxes. All methods are selected by default with the report displaying values against only the selected methods.
- Select which issuers to run the report for. **All Issuers** is selected by default. To run the report for one or more Issuer Groups and/or Issuers, deselect the **All Issuers** checkbox and use the **Add>>**, **<<Remove** buttons to select **Issuers** or **Issuer Groups**.

Warning

Run the report for **All Issuers** with caution as it may take a significant time to produce the report.

- **Extend report by device type** by selecting the checkbox. If selected, at least one device should be selected to run the report.

- **3-D Secure Version** by selecting the **3DS1** and / or **3DS2** checkboxes, as appropriate.
- Click the ***Go*** button to display the new report.
- Click the ***Export*** button in order to export the currently displayed report as a Comma Separated Value (CSV) file.
- Click the ***Back*** button to modify the search criteria.

Enrolment Activity

Reports > Enrolment Activity

The enrolment report shows the total number of enrolled, pre-registered and cancelled cards for a given period. The report is also broken down by the authentication method.

The default report shows monthly card enrolments for the current year for all issuers. You may select any number of issuers and/or issuer groups, a different daily, monthly or annual report; and specify a period.

The **Enrolled Cards** column shows the total number of cards enrolled (fully registered) in the selected period. Some of these cards may have been cancelled, which appear in the **Cancelled Cards** column. The difference between Enrolled Cards and Cancelled cards is the number of enrolled cards remaining.

The **Pre-registered** cards column shows the number of cards, which have been pre-enrolled by the banks for those cardholders who have not yet finalised their registration. Some of these cards may be cancelled before the cardholder has finalised their enrolment. Currently the system does not log this event and the statistics for cancelled pre-registered cards is not available.

Use the following fields to produce an enrolment report:

- Select **Monthly**, **Daily** or **Annual Period** from the drop down list.
- Enter a date range in the **From** and **To** dates (dd/mm/yyyy). Defaults are: **From: 01/** and **To: 12/**.
- Select at least one **Provider**, by selecting/deselecting the appropriate checkboxes. All providers are selected by default with the report displaying values against only the selected Providers.

- Select which issuers to run the report for. **All Issuers** is selected by default. To run the report for one or more Issuer Groups and/or Issuers, deselect the **All Issuers** checkbox and use the **Add>>**, **<<Remove** buttons to select **Issuers** or **Issuer Groups.**

> **Warning**
>
> Run the report for **All Issuers** with caution as it may take a significant time to produce the report.

- Click the ***Go*** button to display the new report.
- Click the ***Export*** button in order to export the currently displayed report as a Comma Separated Value (CSV) file.
- Click the ***Back*** button to modify the search criteria.

Merchant Activity

Reports > Merchant Activity

The merchant activity report shows the total number of authentications initiated by top merchants for a given period. The report is also broken down by the authentication method and successful and failed authentications.

By default, the report shows top 10 merchants' activity for the current year based on total authentication requests send by the merchant. You may select a different period or view top 20 or 50 merchants instead. You may also select the report to be generated and sorted based on the authentication scheme or the number of successful and failed authentications.

Use the following fields to produce a merchant activity report:

- Select Top 10 Merchants (default), Top 20 Merchants or Top 50 Merchants from the **Show** drop down list
- Select the authentication type from the **Based on** drop down list. Defaults to **Total Authentications**.
- Enter a date range in the **From** and **To** dates (dd/mm/yyyy). Defaults are: **From: 01/01/** and **To: 31/12/**.
- Select at least one **Authentication method**, by selecting/deselecting the appropriate checkboxes. All methods are selected by default with the report displaying values against only the selected methods.

- Select which issuers to run the report for. **All Issuers** is selected by default. To run the report for one or more Issuer Groups and/or Issuers, deselect the **All Issuers** checkbox and use the **Add>>**, **<<Remove** buttons to select **Issuers** or **Issuer Groups**.

> **Warning**
>
> Run the report for **All Issuers** with caution as it may take a significant time to produce the report.

- **Extend report by device type** by selecting the checkbox. If selected, at least one device should be selected to run the report.
- **3-D Secure Version** by selecting the **3DS1** and / or **3DS2** checkboxes, as appropriate.
- Click the ***Go*** button to display the new report.
- Click the ***Export*** button in order to export the currently displayed report as a Comma Separated Value (CSV) file.
- Click the ***Back*** button to modify the search criteria.

Purchases

Reports > Purchases

The purchase report shows the total purchase volume for a given period. The report is also broken down by the authentication method. The purchase volume is divided based on the purchase currency.

The report shows monthly purchase volume for the current year by default. You may select a different daily, monthly or annual report and specify a period or choose a currency for the report.

> **Note**
>
> Please note that authentication requests display the currency in which the transaction will be cleared by the merchant and do not specify any international exchange rates involved in the authorization process. As such the purchase report may have to be specified in multiple currencies.

Use the following fields to produce a purchases report:

- Select a **Currency** or **All** from the drop down list
- Select **Monthly**, **Daily** or **Annual Period** from the drop down list

- Enter a date range in the **From** and **To** dates (dd/mm/yyyy). Defaults are: **From: 01/01/** and **To: 31/12/**.
- Select at least one **Authentication method**, by selecting/deselecting the appropriate checkboxes. All methods are selected by default with the report displaying values against only the selected methods.
- Select which issuers to run the report for. **All Issuers** is selected by default. To run the report for one or more Issuer Groups and/or Issuers, deselect the **All Issuers** checkbox and use the **Add >>**, **<<Remove** buttons to select **Issuers** or **Issuer Groups**.

> **Warning**
>
> Run the report for **All Issuers** with caution as it may take a significant time to produce the report.

- **Extend report by device type** by selecting the checkbox. If selected, at least one device should be selected to run the report.
- **3-D Secure Version** by selecting the **3DS1** and / or **3DS2** checkboxes, as appropriate.
- Click the ***Go*** button to display the new report.
- Click the ***Export*** button in order to export the currently displayed report as a Comma Separated Value (CSV) file.
- Click the ***Back*** button to modify the search criteria.

The following are displayed by the Period specified:

- Period - days, months or years
- Names of the issuer groups / issuers selected or 'for all issuers' if the **All Issuers** checkbox was selected
- Number of Transactions
- ***Total Amount for transactions by transaction currency***
- SecureCode Transactions
- SecureCode Total
- VbV Transactions
- VbV Total
- JCB J/Secure Transactions
- J/Secure Total

- American Express SafeKey Transactions
- SafeKey Total
- Diners Club International ProtectBuy Transactions
- ProtectBuy Total

Admin

Reports > Admin

This report provides a summary of administrative user accounts by issuer. The report is broken down by user access type and can be customised to include one or more issuers or issuer groups.

Use the following fields to produce an admin report:

- Select which issuers to run the report for. **All Issuers** is selected by default. To run the report for one or more Issuer Groups and/or Issuers, deselect the **All Issuers** checkbox and use the **Add>>**, **<<Remove** buttons to select **Issuers** or **Issuer Groups.**
- Click the ***Go*** button to display the new report.

The admin report is broken down into three sections: summary, admin users per issuer and admin users per group. Summary shows the total number of admin users across the system based on their access type and available to system users only. Issuer and group based reported show admin users per issuer and group, respectively.

- Click the **Export** link to export the currently displayed report as a Comma Separated Value (CSV) file.
- Click the **Back** link to modify the search criteria.

Audit Log

System Administrators, Issuer Administrators, IT Security Administrators

This section is used for keeping a record of all critical actions performed by administrative users. It has the follow menu options:

Find Audit Log

This section is used to locate and view audit logs. You can search for an audit log by Date Range, Username, User ID, Issuer or Issuer Group, Access type, Event type and Table.

Audit Log > Find Audit Log

Use the following field to search for an audit log:

You can leave all fields at default or blank to display a list of all logs.

- Select a **Target database** from the list to search for **Current** or **Archived** audit logs.
- Enter a date range in dd/mm/yyyy format in the **From** and **To** fields.
- Enter all or part of the **Username**
- Enter a **User ID**, which is a unique ID assigned to each user when first created. Unlike username, user ID remains unchanged and can be used to identify a user, in case the username has changed.
- **Access type** defaults to **All.** Deselect the **Access type** checkbox to select from the drop down list. Select multiple Access types using Ctrl+click.
- When **Access Type** is set to **Event** or **All**, **Event type** defaults to All. Deselect the **Event type** checkbox to select from the drop down list. Select multiple Event types using Ctrl+click.
- For all Access types, other than **Event**, you can optionally select a database **Table** which limits the results to actions performed on the selected table.

Tables defaults to **All.** Deselect the **Tables** checkbox to select from the **Available** list and then click the **Add>>** button to transfer the **Table** to the **Selected** list. Select multiple Tables using Ctrl+click.

- ***Search*** button to display results.

Search Result

This page displays logs matching your search criteria.

Use the *No* or *Date* links to view details for a log.

Audit Log > Find Audit Log > Search Result

Fields & links displayed on this page:

- **No** - records are numbered for reference purposes only for each search performed - this field links to **Audit Log Details** page for selected record
- **Date** link to **Audit Log Details** page for selected record.
- **Time** log was recorded
- Database **Table** accessed
- Type of **Access** - Event, Update, Insert or Delete
- **Username**
- **Issuer**
- **Group**

Log Details

This page displays full details for the audit log record selected on the **Search Result** page.

Audit Log > Find Audit Log > Search Result > Log Details

Fields displayed on this page:

- **Access ID**
- **Access date**
- **Username**
- **User ID**

- **Type of access**
- **Description**
- **Client IP**
- **Object name**
- **Issuer**
- **Group**

Fields displayed for the database table changed:

- **Field**
- **Old Value**
- **New Value**

Profile Management

All Admin Users can edit their profile and change their password.

It is recommended that you change your password on a regular basis for security reasons or if you suspect that security has been compromised by another user logging in with your username and password.

The Change Password function is accessed via the **Edit Profile** link displayed on the right of the title bar area. You can also use this link to keep your contact details up to date.

- Click the Edit Profile hyperlink

  The **Edit Profile** page is displayed.

Edit Profile

Use the following fields to change your details or password:

- The **Username** of the user currently logged on is displayed and cannot be changed.
- User Details
- Enter your **Full name**.
- You must enter a valid **email address**
- Enter your **Contact number**.
- Enter your **Address**.

  Password Details
- Enter your current password as **Old password**.
- Enter the **New Password** you have chosen.

> **Warning**
>
> Always choose a password that you have not used for the Administration Server previously. The Administration System keeps a history of the last 10 passwords and does not allow you to reuse passwords in the history. For example, you cannot keep two favourite passwords and rotate them. If you try to reuse a password stored in the history a message is displayed: **This password has been selected before**.

- Re-enter your new password as **Re-enter Password**.
- **Two-factor authentication login** checkbox

  Select this checkbox if you want to enable two-factor authentication when logging in.

> **Info**
>
> By selecting the checkbox, a **QR code** and a **Secret key** are displayed. You can either scan the QR code or enter the Secret key manually in Google Authenticator to receive the Authenticator Code. Two-factor authentication will be enabled once you enter the Authenticator Code. For more information, refer to Login.

- Click the **Apply** button to save your details.

About Setup Guide

This section is a post installation guide for setting up ActiveAccess for testing. It describes, at a high level, the steps involved in setting up issuers, signing certificates and configuring cards for various payment scenarios, prior to testing. It also covers the steps involved in device configuration, setting up remote / external authentication, archiving and RBA. It should be read in conjunction with Administration UI, which provides additional information on completing each step.

Document Conventions

The following colours are used to indicate in which environment setup steps are to be performed.

Colour	Environment
	ActiveAccess Administration
	ActiveMerchant Test Payment Page (GPayments MPI (ActiveMerchant)) or an equivalent third-party MPI
	GPayments Licensing
	Certificate Authority

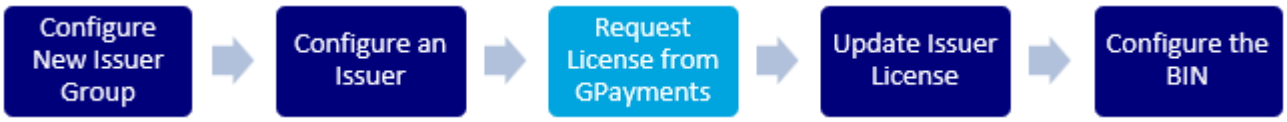
ACS URL

This section covers configurations related to *3DS1 only*.

System Management > ACS Settings > Authentication server: Local or Remote

1. Enter the **ACS URL** (e.g. https://yourserverip:port/acs/pa)
2. Click the **Apply** button.

Issuer Groups and Issuers



Configure a New Issuer Group (Optional)

System Management > Group Management

- Check for an Issuer Group relevant to the client by looking under the **Group Name** column.
- If the required Issuer Group exists, go to Configure an Issuer, otherwise click the *New Issuer Group* link
- Enter the Issuer Group **Name** (*e.g.* Company Issuer Group)
- You may skip the remaining fields or fill them, as appropriate
- Click the **Apply** button.

> **Note**
>
> Cryptographic keys are created for the issuer signing certificate and CAVV validation.

Configure an Issuer

System Management > Issuer Management

- Check for a relevant issuer by looking under the **Issuer Name** column or searching for it by **Issuer Name**
- If the issuer exists, go to Section 4.3 - Request and Update Issuer License, otherwise click the *New Issuer* link
- Enter the Issuer **Name** (e.g. Test Issuer, Test Bank)
- Select the Issuer Group, if one was created in Section 4.1 - Configure a New Issuer Group, from the **Parent group** drop down list and tick the checkboxes for **Use parent certificate, public and encryption keys** and **Use parent keys**

- You may skip the remaining fields or fill them, as appropriate
- Click the **Apply** button.

> **Note**
>
> Cryptographic keys are created for encrypting the cardholder and transaction data of the specified issuer.

Request and Update Issuer License

ActiveAccess License

- Contact GPayments and request a license key for the issuer created in Section 4.2 - Configure an Issuer
- Copy the license key provided to you by GPayments to your clipboard.

System Management > Issuer Management

- Find the issuer and click the *Issuer Name*
- On the **Issuer Details** page, paste the copied **License Key** into the text box
- Click the **Apply** button.

> **Note**
>
> If an error occurs, contact GPayments Tech Support.

Configure the BIN

System Management > Issuer Management

- Find the issuer and click the *Issuer Name*
- On the **Issuer Details** page, click the *BIN Management* link
- On the **BIN Management** page, click the *Add BIN* link
- On the **Add BIN** page, enter the **BIN** (e.g. 412345)
- Ensure **Status** is set to **Enabled**
- Select an option from the drop down list for **Device over 3-D Secure**, as appropriate

- Click the **Apply** button.

Certificate Signing Requests

Configure Issuer Group Signing Certificates

Issuer Group Signing Certificates should be configured individually for each provider.

Security > Issuer Certificate

- Click the *Create Certificate Request* link
- On the **Certificate Request** page
  - Select the **Issuer** or **Issuer Group** from the drop down list
  - Select the required **Provider** from the drop down list
  - Fill the remaining fields as appropriate
  - Click the **Apply** button.
- **Certificate Signing Request** (CSR) – Copy the contents of the CSR or click the **Download** button to save the CSR.

Sign the Certificate Signing Request (CSR)

- Sign the Certificate Signing Request (CSR) using a Certificate Authority.

Install the Certificate Request

Security > Issuer Certificate

- Click the *Install Certificate* link
- Select the **Issuer** or **Issuer Group** from the drop down list

- Select the required **Provider** from the drop down list (This must be the same as the provider selected for the Certificate Request in Section 5.1 - Configure Issuer Group Signing Certificates)
- Click the **Choose File** button to locate and select the **Signed Certificate** file or click the **Certificate content** radio button and paste the Signed Certificate content
- Click the **Apply** button
- Ensure that it is completed successfully.

Cards

This section covers configurations related to *3DS1 only*.

Add a New Card

Users > New Card

- Select **Issuer** from the drop down list
- Select the **Authentication method** from the drop down list (This should correspond to the card provider)
- Ensure **Status** is set to **Enabled**
- Enter the **Card number**
- Enter the cardholder name in **Name on card**
- Enter the **Expiry date**
- Set the **Internet PIN.** This will be used during Activation During Shopping when registering the Pre-registered card)
- Click the **Apply** button.

Note

Cards can also be uploaded in bulk through ActiveAccess Registration Requests and the GPayments Card Loader application. Refer to the ActiveAccess documentation, ActiveAccess Administration and GPayments Card Loader and Signer/Verification Application, for further information.

Configure Custom Pages

Upload Local Custom Pages

Issuers > Custom Pages

- Select the **Issuer** or **Issuer Group** radio button and select from the drop down list
- If matches are found, custom pages have previously been set up for this issuer, in which case go to Section 8 - Authentication Scenarios Setup
- If no matches are found, click the *Upload File* link
- Use the **Choose File** button to locate and upload the *Authentication.zip* file from the following path in ActiveAccess installation package: ActiveAccess/data/custompage/issuer/Any Bank

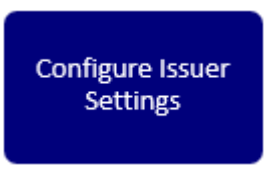
> **Note**
>
> You can customise the XSL pages as appropriate. Note that different custom pages are used for local and remote issuers.

- Click the **Apply** button.

Authentication Scenarios Setup

Each authentication scenario covered in this section should be set up and tested independently, *using a newly created card*, as individual scenarios may require a different configuration within the same issuer.

Activation During Shopping (ADS) Scenario



Configure Issuer Settings

Issuer > Settings

- Select the **Issuer** from the drop down list
- Set **Activation During Shopping** to **Enabled** for all requested/configured card providers
- Click the **Apply** button.

Perform a Test Transaction

- Go to Section 9 - Perform a Test Transaction.

Authentication Success Scenario

Configure Issuer Settings

Issuer > Settings

- Select the **Issuer** from the drop down list
- Set **Activation During Shopping** to **Enabled** for all requested/configured card providers
- Click the **Apply** button.

Register the Pre-Registered Card

Perform a test transaction with the card to register the card through Activation During Shopping (ADS). If you have access to the GPayments MPI (ActiveMerchant), follow the steps below.

ActiveMerchant Test Payment Page

- Go to the ActiveMerchant Test Payment page to register a newly created card set up in Section 6 - Configure Card Numbers
- Enter the **Card number** to perform a test transaction
- Click the **Submit** button
- On the confirmation page, click the **Submit** button

- On the registration page, enter **Name on Card** and **Internet Pin**, which were set up in Section 6 - Configure Card Numbers
- Click the **Submit / Activate** button
- Enter a **Personal Assurance Message**
- Set a **Password** to be used for authenticating the cardholder
- Click the **Submit** button
- On the next page, click the **Continue / OK** button to go to the Success page.

PERFORM A TEST TRANSACTION

- Go to Section 9 - Perform a Test Transaction.

Authentication Fail Scenario

Configure Issuer Settings

Issuer > Settings

- Select **Issuer** from the drop down list
- Set **Activation During Shopping** to **Enabled** for all requested/configured card providers
- Set an appropriate value for **Maximum unsuccessful attempts**. As the card will need to be locked for this scenario (Section 8.3.3 - Lock the Card), it is preferable to set a lower value, e.g. 3. Do not set the value to 0 (disable).
- Set **Automatic unlock** to 0 (disabled)

> **Note**
>
> Perform this step only if you would like the cards of this issuer to stay locked and provide results for the authentication failed scenario each time.

- Click the **Apply** button.

Register the Pre-Registered Card

Perform a test transaction with the card, to register the card through Activation During Shopping (ADS). If you have access to the GPayments MPI (ActiveMerchant), follow the steps below.

ActiveMerchant Test Payment Page

- Go to the ActiveMerchant Test Payment Page to register the card set up in Section 6 - Configure Card Numbers
- Enter the **Card number** to perform a test transaction
- Click the **Submit** button
- On the confirmation page, click the **Submit** button
- On the registration page, enter **Name on Card** and **Internet Pin** which were set in Section 6 - Configure Card Numbers
- Click the **Submit** button
- Enter a **Personal Assurance Message**
- Set a **Password** to be used for authenticating the cardholder
- Click the **Submit** button
- On the next page, click the **Continue / OK** button to go to the Success page.

Lock the Card

To lock the card, perform a test transaction with the card, entering an incorrect password until the card is locked. If you have access to the GPayments MPI (ActiveMerchant), follow the steps below.

ActiveMerchant Test Payment Page

- Go to the ActiveMerchant Test Payment page
- Enter the **Card number** to perform a test transaction
- Click the **Submit** button
- On the confirmation page, click the **Submit** button
- On the authentication page, enter an incorrect password. Repeat, until the message *This Account is Locked!* is displayed.
- Click the **OK** button.

Perform a Test Transaction

- Go to Section 9 - Perform a Test Transaction.

Forgot Password Scenario

Configure Issuer Settings

System Management > Issuer Management

- Find the issuer and click the *Issuer Name* to go to the **Issuer Details** page
- Set **Show extended account information** to **Yes** for Question and Answer fields be shown on **Card Details** page
- Click the **Apply** button.

Issuer > Settings

- Select the **Issuer** from the drop down list
- Set **Activation During Shopping** to **Enabled** for all requested/configured card providers
- Click the **Apply** button.

Register the Pre-Registered Card

To register the card through Activation During Shopping (ADS), perform a test transaction with the card. If you have access to the GPayments MPI (ActiveMerchant), follow the steps below.

ActiveMerchant Test Payment Page

- Go to the ActiveMerchant Test Payment page to register the card set up in Section 6 - Configure Card Numbers
- Enter the **Card number** to perform a test transaction
- Click the **Submit** button
- On the confirmation page, click the **Submit** button

- On the registration page, enter **Name on Card** and **Internet Pin** which were set in Section 6 - Configure Card Numbers
- Click the **Submit** button
- Enter a **Personal Assurance Message**
- Set a **Password** to be used for authenticating the cardholder
- Click the **Submit** button
- On the next page, click the **Continue / OK** button to go to the Success page.

Configure Question & Answer

Users > Find Card > Card Details

- Enter a **Question** and an **Answer**
- Click the **Apply** button.

Perform a Test Transaction

- Go to Section 9 - Perform a Test Transaction. During the transaction, click on the "Forgot your Password?" link.

Proof of Attempt Scenario

Configure Issuer Settings

Issuer > Settings

- Select **Issuer** from the drop down list
- Set **Activation During Shopping** to **Disabled** for all requested/configured card providers
- Set **Proof of Authentication Attempt** to **Enabled** for all requested/configured card providers
- Click the **Apply** button.

Perform a Test Transaction

- Go to Section 9 - Perform a Test Transaction.

PAN Not Enrolled Scenario

Configure Issuer Settings

Issuers > Settings

- Select **Issuer** from the drop down list
- Set **Activation During Shopping** to **Disabled** for all requested/configured card providers
- Set **Proof of Authentication Attempt** to **Disabled** for all requested/configured card providers
- Click the **Apply** button.

Perform a Test Transaction

- Go to Section 9 - Perform a Test Transaction.

Delay / Timeout Scenario

You can test scenarios such as delay or timeout in Verify Enrolment or Payer Authentication processes.

You can create a **responseTimeout.properties** file in ActiveAccess' **AA_HOME** directory. This configuration file can be used for testing purposes only and under no circumstances should be used in a real production environment. The properties in this configuration file should be in the following general format:

- **BIN.reqType**=waiting time (milliseconds)
- **BIN**: is the issuer specified BIN number for which you would like to cause a delay in the response to card numbers that match the BIN.
- **reqType (VE or PA)**: is the type of the request for which you would like to cause a given amount of delay. VE and PA stand for Verify Enrolment and Payer Authentication requests respectively.
- **Waiting time**: The delay in milliseconds that you would like to cause in the response of VE or PA requests.

To set a VE response delay, it is recommended that it is greater than the VERES time-out defined by the MPI.

To set PA response delay, it is recommended that it is greater than the PARes time-out defined by the MPI

Info

0 means unlimited

Example

412345.VE=15000

512345.PA=20000

341234.VE=0

ActiveAccess server should be restarted for changes to take effect.

Perform a Test Transaction

After configuring each of the authentication scenarios, perform a test transaction with the card. If you have access to the GPayments MPI (ActiveMerchant), follow the steps below.

ActiveMerchant Test Payment Page (3-D Secure 1)

- Go to the ActiveMerchant Test Payment page
- Enter the **Card number** to perform a test transaction
- Click the **Submit** button
- Ensure that the outcome corresponds with the authentication scenario.

Devices

This section covers the configuration of licenses and BINs to enable the use of devices for authentication. Examples of some common devices have been included below.

Note that when device configuration is complete, devices can be assigned to individual cards during transactions or via ActiveAccess Administration in Users > Find Cards > Card Details > Assigned Devices > Device Management.

Configure License and BINs

Request and Update License

For issuers to be device compatible, a license needs to be issued with ActiveDevice support.

ActiveAccess License

- Contact GPayments and request for a license key with ActiveDevice support for the issuer
- Copy the license key provided to you by GPayments to your clipboard

System Management > Issuer Management

- Find the issuer and click the *Issuer Name*
- On the **Issuer Details** page, paste the copied **License Key** into the text box
- Click the **Apply** button.

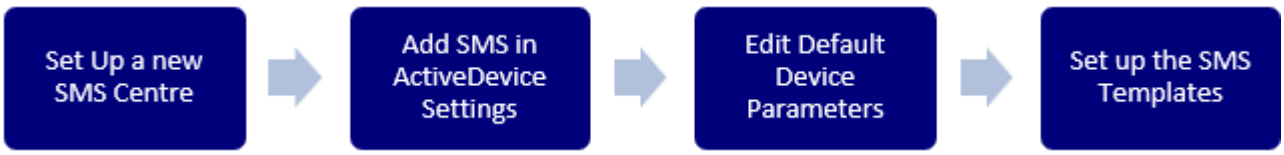
> **Note**
>
> If an error occurs, contact GPayments Tech Support.

Enable Device over 3-D Secure for BINs

System Management > Issuer Management > Issuer Details > BIN Management

- If the BIN has been created previously and has **Device over 3-D Secure** set to **Disabled**, follow the steps below:
  - Click on the *BIN* to go to the **BIN Details** page
  - Set **Device over 3-D Secure** to **Enabled**
  - Click the **Apply** button.
- If new BINs are being created, follow the steps below:
  - Click the *Add BIN* link
  - On the **Add BIN** page, enter the **BIN** (e.g. 412345)
  - Ensure **Status** is set to **Enabled**
  - Set **Device over 3-D Secure** to **Enabled**
  - Click the **Apply** button.

SMS



Set up a New SMS Centre

System Management > Device Management > Edit Default Device Parameters > Device Type: SMS > SMS Centres > New SMS Centre

- Enter a **Name** for the SMS centre
- Enter the **Domain/IP** of the SMS centre
- Enter the **Port** number of the SMS centre
- Enter **System ID**, **System type** and **Password**, if required
- Enter **Sender's mobile number**
- Click the **Apply** button.

Add SMS in ActiveDevice Settings

System Management > Issuer Management > Issuer Details > ActiveDevice Settings

- Under **Supported devices**, in the **Available** box, select SMS and click the **Add >>** button.
- Click the **Apply** button.

Edit Default Device Parameters

System Management > Device Management > Edit Default Device Parameters

- Select **SMS** from the **Device type** drop down list
- Update the device parameters as appropriate
- Click the **Apply** button.

Email

Add Email in ActiveDevice Settings

System Management > Issuer Management > Issuer Details > ActiveDevice Settings

- Under **Supported devices**, in the **Available** box, select Email and click the **Add >>** button.
- Click the **Apply** button.

Edit Device Parameters

System Management > Issuer Management > Issuer Details > ActiveDevice Settings > Device parameters

- Select **Email** from the **Device type** drop down list
- Untick **Use device's default parameters**
- Update the device parameters as appropriate
- Click the **Apply** button.

Set up Email Templates

System Management > Issuer Management > Issuer Details > ActiveDevice Settings > Device parameters > Device Type: Email > Email Templates

- Select a **Template name** from the drop down list
- Adjust the template as required using the **Template** textbox and check the **Preview** textbox for **Plain Context type** or click *Send Test Email* for **HTML Context type**.
- Click the **Apply** button.

> **Note**
>
> The settings and templates configured in **10.3.2 Edit Device Parameters** and **10.3.3 Set up Email Templates** will apply to the specific issuer only. To set default device parameters and templates that apply to all issuers, go to **System Management > Device Management > Edit Default Device Parameters**. The default configurations will apply to all issuers, unless **Use device's default parameters** is unticked in the issuer's device configurations.

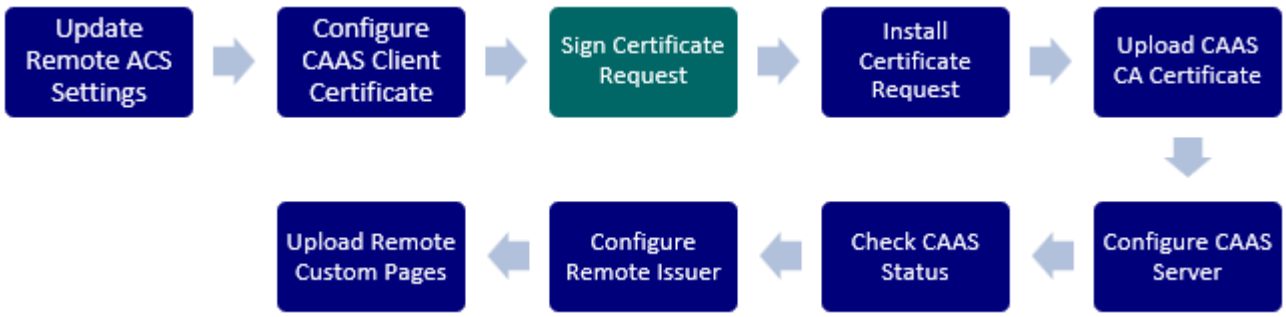
VASCO

Upload the VASCO File

System Management > Device Management > Upload File

- Select the Issuer from the **Issuer** drop down list
- Select **VASCO** from the **Device type** drop down list
- Click the **Choose File** button to locate and select the appropriate VASCO file
- Enter the **Key value**
- Set up a **Schedule** as required
- Click the **Apply** button.

Remote/External Authentication



- Before commencing remote authentication setup, make sure that the required Web services have been implemented and that the CAAS server is up and running.

Update Remote ACS Settings

System Management > ACS Settings > Authentication server: Remote (CAAS)

- Enter the **ACS URL** (e.g. https://yourserverip:port/acs/pa)
- Click the **Apply** button.

Configure CAAS Client Certificate

Security > CAAS Certificate

- Click the *Create Certificate Request* link
- On the **CAAS Certificate Request** page:
  - Enter the certificate details, as appropriate
  - Click the **Apply** button.
- Certificate Request – Copy the certificate contents or click the **Download** button to save the certificate request.

Sign Certificate Request

- Sign the Certificate Request using a Certificate Authority.

Install Certificate Request

Security > CAAS Certificate

- Click the *Install Certificate* link
- Use the **Choose File** button to locate and select the **Signed Certificate** file or click the **Certificate content** radio button and paste the Signed Certificate content.
- Click the **Apply** button.

Upload CAAS CA Certificate

Security > CA Certificate

- Click the *Import CA Certificate* link
- Select **CAAS client** from the **Provider** drop down list
- Click the **Choose File** button to locate and select the **CA Certificate** file
- Click the **Import** button.

Configure CAAS Server

Servers > CAAS Servers

- Click the *Add CAAS Server* link
- On the **Add CAAS Server** page:
  - Enter **CAAS URL** of the CAAS server
  - Enter a value for **CAAS Connection timeout**
  - Enter a value for **Maximum SMS request**
  - Fill the remaining fields as appropriate
  - Click the **Add** button.

Check CAAS Status

Servers > CAAS Servers

- Click the *CAAS URL* link
- On the **Edit CAAS Server** page, click the *Check CAAS Status* link

- On the **Check CAAS Status** page, ensure the message displayed indicates that CAAS is up and running.

Configure Remote Issuer

System Management > Issuer Management

- Find the issuer and click the *Issuer Name*
- If the issuer does not exist, refer to Section 4 - Issuer Group and Issuer Setup to configure issuer, license and BIN and then continue with the next step, otherwise click the *Issuer Name* link
- On the **Issuer Details** page:
  - Set **Authentication server** to **Remote (CAAS)**
  - Select the URL of the **CAAS Server** from the drop down list
  - Click the **Apply** button.

Upload Remote Custom Pages

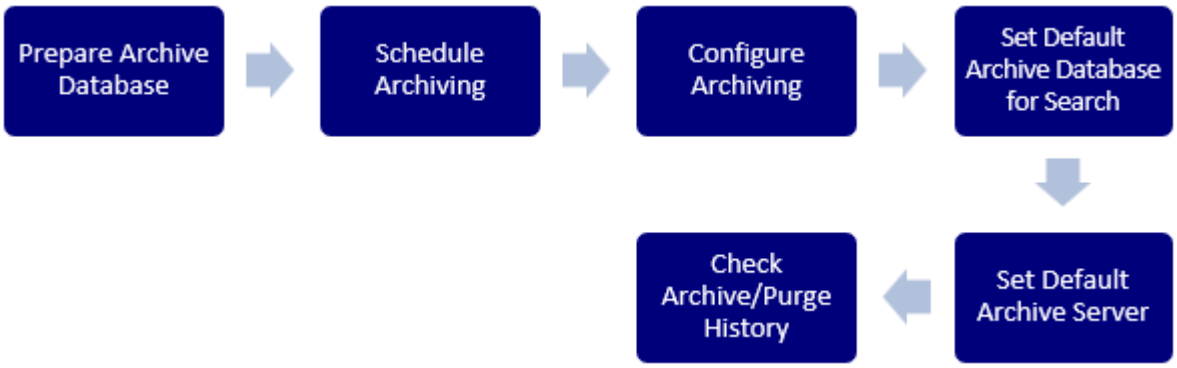
Issuers > Custom Pages

- Select the **Issuer** or **Group** radio button and select from the drop down list
- Click the *Upload File* link
- Use the **Choose File** button to locate and upload the *Authentication.zip* file from the following path in ActiveAccess installation package: ActiveAccess/data/custompage/issuer/AnyBank_Remote

> **Note**
>
> You can customise the XSL pages as appropriate. Note that custom pages are different for local and remote issuers.

- Click the **Apply** button.

Database Archiving



Prepare Archive Database

- Create a new database user with the appropriate permissions
- Connect to the database user and run *archive_schema.sql* from the **Archive** folder in ActiveAccess installation package.
- To give access to the current ActiveAccess database user
- In the *archive_grant.sql* from the **Archive** folder in ActiveAccess installation package, replace the tags **< archiveusername >** with the newly created database user for archiving, **< dbname >** with the database owner name, and **< dbuser >** with the database user name that accesses the database. In a simple configuration the database user name may be the same as the database owner name.

> **Note**
>
> The < dbname > and < dbuser > can be found in **AA_HOME/activeaccess.properties** as **DBNAME** and **DBUSERNAME** respectively.

- Run the updated **archive_grant.sql** with a sys/system connection.

Schedule Archiving

System Management > Archive Management

- Click the *Edit* link
- Tick the **Automatic archive** checkbox to enable automatic archiving
- Fill the remaining fields as appropriate

- For purging the archived data:
  - Tick the **Automatic archive purge** checkbox to enable automatic archiving
  - Fill the remaining fields as appropriate
- Click the **Apply** button.

Configure Archiving

System Management > Archive Management > Archive Databases > New Archive Database

- Enter the **Archive Database link** or **Database user**
- Click the **Apply** button.

Set Default Archive Database for Search

System Management > Archive Management > Archive Databases

- If you only have one archive database configured, it will automatically be set as the default for search. If you have more than one archive database configured, click the *Set as default for search* link for the desired archive database.

Set Default Archive Server

Servers > MIA Servers

- Click the *Set as default archive server* link for the desired MIA Server.

Check Archive/Purge History

System Management > Archive Management > Archive Databases

- Click the *Archive database* link to go to the **Archive Database Details** page

  A list of archive and purge history reports will appear under **Archive History** and **Purge History** tabs

Local Messaging

> *Previously AA32-GPayments Card Loader.pdf*

This section details the messaging specification for the ActiveAccess cardholder and user registration application programming interface (API).

Cardholder Registration

The authentication system is responsible for authentication of cardholders during American Express SafeKey, Diners Club International ProtectBuy, JCB J/Secure, Mastercard SecureCode / IDC and Verified by Visa / Visa Secure transactions. As a result, it is necessary that the system stores cardholder related information to the extent that this requirement can be satisfied. Cardholder registration is the process through which an issuer registers this information with ActiveAccess. Depending on the choice of registration model, cardholders may or may not have to complete an enrolment process. Please note, the distinction between 'registration' and 'enrolment,' as used in this document, where registration is carried out by the issuer (3DS1 and 3DS2) and enrolment is performed by the cardholder (3DS1 only).

There are three cardholder registration models for populating data into the authentication system. **Direct entry** (3DS1 only) allows help desk operators to manually enter cardholder registration data through the ActiveAccess administration interface. The **pre-registration** (3DS1 only) and **final registration** models both format cardholder data in an XML message. These messages are either sent directly to the registration server via an API, using the CardLoader application, or uploaded through the ActiveAccess Administration interface. To facilitate these processes and integrate with the authentication system some integration work with the issuer's systems is required.

Info

Refer to New Card for more information on the direct entry cardholder registration model.

The purpose of registration is to upload or supply the information necessary to enable a cardholder to use the authentication protocols. For each registered card, the registration message must include the required fields; card type, card number and card name.

There are four types of XML messages accepted by the system for the purpose of cardholder creation and maintenance. They are:

- **PreReg** (3DS1 only): enables the creation of pre-registered cardholder data in the system. When cards have this status, there is no authentication data available for the card. Authentication data can be assigned to the card by:
  - the cardholder through the Enrolment component (supports static password only)
  - the cardholder via Activation During Shopping
  - admins via MIA > Cards
  - admins via uploading of FinalReg requests.
- **FinalReg**: enables the creation of fully registered cardholder data in the system. When cards have this status, they have one or more authentication data that can be used for authentication.
- **UpdateReg**: enables the alteration of cardholder data in the system
- **CancelReg**: enables the removal of cardholder data from the system.

> **Info**
>
> **Section - Request** provides more detail on the format of these messages.

To perform authenticated transactions, it is essential that the cardholder data elements stored within the authentication system is sufficient to allow this. The essential elements required are:

- Card type
- Card number
- Card name
- Personal assurance message (personal greeting)

Enrolled cards may also use the following elements:

- Password (J/Secure Password, ProtectBuy Password, Mastercard SecureCode, SafeKey Password, or Visa Password).
- Expiry Date
- Device Type
- Device Serial Number

- Hint (used for authorisation retry prompt)
- Hint and Response (question and answer pair used to verify cardholders if they have forgotten their authentication password)

The information required for cardholder authentication is primarily provided to the authentication system by the card issuing member bank. The Registration API provides a flexible data transport and definition mechanism that allows each member bank to build an individual cardholder registration process to meet their specific requirements.

Alternatively, cardholder data can be formatted into XML files and uploaded through the ActiveAccess administration interface. This process reduces the technical requirements for the issuer.

Pre-registration (3DS1 only)

When using the pre-registration model, a member bank only needs to establish the authentication criteria for enrolment of cardholders and provide this information to the authentication system. The actual enrolment process is then handled by the authentication system's enrolment module. The authentication system uses the pre-registration information to verify the identity of the cardholder and set-up the cardholder account during the cardholder enrolment process.

In the pre-registration model, it is essential that a **PreReg** message uploads the card type, card number and card name, regardless of the protocol. It is also required that the issuer supplies a number of other known parameters to the authentication system to verify the cardholder when they come to enrol with the authentication system. The following provides a list of suggested data that can be included in the **PreReg** message to allow the cardholder authentication to occur. Please note that these fields are at the discretion of the issuing bank:

- Date of birth
- Mother's maiden name
- Card verification check number
- Credit limit
- Billing address

An example of pre-registration is when the member bank mails invitations to cardholders and provides them with, say, a registration number. Cardholders can then visit the authentication system's online enrolment website or enrol using the activation during shopping process. Having

entered their card number, card name and registration number and once the cardholder's identity is successfully verified, they can proceed with setting up their account, which includes selection of an American Express SafeKey, Diners Club International ProtectBuy, JCB J/Secure, Mastercard SecureCode or Verified by Visa password. This static or dynamic authentication data will then be used in all subsequent authenticated transactions.

Final Registration

The final registration model allows issuers to control the cardholder enrolment process. It requires the member bank to develop its own enrolment process to collect cardholder enrolment information. Once the cardholder enrolment process is complete, the information is sent to the authentication system in the form of a final registration message.

There are two types of final registration, final registration for traditional 3-D Secure authentication and final registration for two-factor authentication over 3-D Secure.

In the final registration model for traditional 3-D Secure, it is essential that a **FinalReg** message uploads the card type, card number and card name, regardless of the protocol. It is also required that the issuer supplies the cardholder authentication fields:

- SafeKey, J/Secure Password, ProtectBuy Password, Mastercard Identity Check / SecureCode, or Visa Password
- Personal Assurance Message

In the final registration model for two-factor authentication over 3-D Secure it is also essential that the issuer supplies the device information which will be used for generating tokens at authentication time in addition to above fields:

- Device Type
- Device Serial Number

An example of final registration is enrolment of cardholders through the issuer's Internet banking site. A possible scenario is for the cardholder to log into the Internet banking site and enable American Express SafeKey, Diners Club International ProtectBuy, JCB J/Secure, Mastercard SecureCode or Verified by Visa for their credit card. The cardholder will then be prompted to choose a SafeKey, J/Secure password, ProtectBuy Password, Mastercard SecureCode or a Verified by Visa password and personal assurance message for their card. If the issuer wants to enable two-factor authentication over 3-D Secure for the card then the cardholder will be prompted to select a device from a list of supported device types and enter its serial number.

The issuer then formats a final registration message, which is sent to the authentication system in order to complete cardholder enrolment.

The format of the registration request is the same for real-time and batch uploads, except that the batch upload may contain multiple blocks of cardholder data.

The registration API uses XML as the message format and HTTP as the message transport. API calls (requests) can be made to the registration server of the authentication system. Issuers can use the registration API to automate their cardholder registration process.

XML Message Format

XML messages must be produced in accordance with the Messaging Requirements Cardholder Registration DTD, as specified in Cardholder Registration DTD.

All request and response messages are enclosed by the `<Message>` element. A message can contain either a **Request and a Signature** or a **Response**.

```
<?xml version="1.0"?>

<Message>

<Request Id="request1" IssuerId="1234567890123456">

<!--the request content here-->

</request>

<Signature xmlns="http://www.w3.org/2000/09/xmldsig#">

<!--the request signature here-->

</Signature>

</Message>
```

<Message>		
Attributes	**Description**	**Usage**
<Request>	Used in the request message, which is sent by the issuer to register one or more cards.	Required for request messages

<Message>		
<Signature>	Issuer signature is used in the request message to prove the identity of the issuer and to validate that the message content has not been altered.	Required for request message
<Response>	Cardholder registration response sent back in response to a cardholder registration request.	Required for response message

Note

The XML response starts with the XML declaration `(<?xml version="1.0" encoding="UTF-8"?>)`. However a request does not need to start with the XML declaration. Request content must be sent with UTF-8 encoding.

Request

A Request is sent by the issuer to perform various cardholder registration tasks. A request can be sent to perform pre-registration, final registration, cancel registration or update registration for one or more cards. A request may only contain one <PreReg>, <FinalReg>, <CancelReg> or <UpdateReg>.

<Request>		
Attributes	**Description**	**Usage**
Id	An arbitrary identifier, which the issuer defines and can be used to refer to the request element as part of a standard URI. Also referenced by the Signature element. XML signature requires the element, which is being signed, to be identified by a unique Id. The value entered should start with an alphabetic character.	Required Max 28 char
IssuerId	A unique identifier for the issuer. Created when the issuer first signs up with the system and supplied during the issuer registration process. Typically an 18 digit numeric value.	Either IssuerId or GroupId is required
GroupId	A unique identifier for the group of issuers. Created when the group is introduced to the system and supplied during the registration process. Typically an 18 digit numeric value. Using a GroupId, cards can be registered for different issuer members within the group in an identical request.	Either IssuerId or GroupId is required

<Request>		
EncVectorIV	If an encryption keystore has been defined for the issuer, or a group of issuers, critical card data must be encrypted using it. The critical data is encrypted using AES/CBC/PKCS5Padding mode, which requires an IV, including 16 random bytes, as an input parameter for encryption and decryption. The IV should be sent to the server to indicate that the card data in the response should be encrypted in CBC mode. To do this, the IV itself must be encrypted in AES/ECB/PKCS5Padding mode, using an encryption key, then base64 encoded and set as EncVectorIV in the request.	Optional, if present, it means that the client has generated an IV parameter and critical card information has been encrypted using the CBC mode and the generated IV, otherwise ECB or plain mode has been used instead.
Elements	**Description**	**Usage**
<PreReg>	Used for pre-registration of one or more cards. Pre-registered Cardholders will need to go through the enrolment process to finalise their registration.	Required for a pre-registration request
<FinalReg>	Used for registration of one or more cards. Registered Cardholders can start making authenticated transactions without the need to go through the enrolment process.	Required for a final registration request
<CancelReg>	Used to remove one or more cards from the system.	Required for a cancel registration request
<UpdateReg>	Used to update the information for one or more cards. For example to change the card number. Cardholder registration status is left unchanged.	Required for an update registration request

Response

A response message is sent back for each request. The response message provides the result of the request message with details of errors, if any. Issuers must process the response message and should correct and replay their request if there is an error.

<Response>		
Attributes	**Description**	**Usage**

<Response>		
EncVectorIV	If an encryption keystore has been defined for the issuer or group of issuers, depending on the encryption key algorithm, the server decrypts critical card data using either AES/ECB/PKCS5Padding or DESede/ECB/PKCS5Padding mode. If the EncVectorIV attribute is set to protect the data using the AES/CBC/PKCS5Padding or DESede/CBC/PKCS5Padding mode, card data is decrypted using the CBC encryption mode and the IV is sent as an input parameter. During processing on the server side, a new random IV is generated and used to encrypt critical card data in AES/CBC/PKCS5Padding or DESede/CBC/PKCS5Padding mode. When the process is complete, the IV itself is encrypted in AES/ECB/PKCS5Padding or DESede/ECB/PKCS5Padding mode using the same key depending on the key algorithm, and then base64 encoded and set as EncVectorIV in the response.	Required, if the client has set this attribute for the request. Server generates a new IV parameter and encrypts critical card information in the response using CBC mode and the new IV, otherwise no attribute will be set and ECB or plain mode will be used instead.
Elements	**Description**	**Usage**
<Code>	Response code. 0 if the request was successful. 1 if the request has been successfully processed but there are warnings. Any other value denotes an error in processing the request.	Required
<ErrorMessage>	A descriptive message that identifies the category of the error.	Required
<ErrorDetail>	A more detailed description of the error.	Required
<Warning>	A warning is issued to provide information on an unexpected situation that does not prevent the request from being successfully processed.	Conditional. Required only if response code is 1.

Note

A message with response code **1** denotes that the request has been successfully processed but yet the registration server has not been able to comply with certain instructions. For example a cancel registration attempt on an already cancelled card is not processed and as such a warning message is reported. An issuer does not need to take any further action in this case.
Warning messages may be logged at the issuer's end for later reference.

> **Note**
>
> Multiple warning messages may be included in a single response message.

Requests

Pre-registration Request

The pre-registration request is typically used by the members who wish to leave cardholder enrolment to the enrolment module. A pre-registered card cannot be used for authenticated transactions. Cardholders are required to finalise their registration by going through the standard enrolment process, which is offered by the enrolment module or through the cardholder activation during shopping process. Pre-registration data is used to verify the identity of cardholders during the enrolment process.

<PreReg>		
Attributes	**Description**	**Usage**
None	N/A	N/A
Elements	**Description**	**Usage**

<PreReg>		
<DataFormat>	Defines a data type, which can be associated with card data. This issuer-defined field has a maximum field length of 1024 characters. You can define the following attributes with the DataFormat: **Name:** The name as used by the program to refer to this data format. Also used by the data element in order to refer to this particular type, e.g.: pass or password **Label:** A short description to appear before the data elements of this type when displayed to the cardholder, e.g.: SecureCode: **Description:** A longer description to appear after the data elements of this type when displayed to the cardholder (optional), e.g.: Please enter your SecureCode **MaxLen:** The maximum length that can be stored in the data elements of this type (optional). **Type:** Can be set to date, string, number or hidden values. When set to date, number, the application performs type validation for the content of the data elements. If not specified the string type is assumed (no validation). If the data format type is set to hidden, this excludes the data format as an authentication data element. Hidden data types are not displayed to the cardholder and can be used as internal field in the XSL pages per bank to achieve certain customised features. You should not use this type unless specifically advised by GPayments to do so **Format:** Additional formatting information can be set to YYYYMMDD or YYYYMM. This is only meaningful when you have set the data format type to date. **Mask:** Can be set to Yes or No. Determines whether the user input for this data format should be masked or not. Set to Yes for password type fields. **DataMode:** This is an optional attribute. Its value can be **Identity**, **identity**, **Auth**, **auth**, **Extension**, or **extension**. Data which has Identity as DataMode will be displayed on the Registration page and may be displayed on the Forgot Password and Hint/Response pages to identify the cardholder during changing or resetting the password. Data which has Auth as DataMode will be displayed during authentication, including the Authentication and Reactivation pages. Data which has Extension as Datamode will be checked by extensions. If it is not set, the default value will be used. In the PreReg file, the default value is Identity.	Required A valid data format must be defined for each type, which is referenced by a <Data> element.
<Card>	Card related data including card number, name on card, expiry date and issuer defined data. Card numbers up to 19 digits are accepted and the name on card field has a maximum length of 128 characters.	Required At least one <Card> must be present

Note

If a data has a 'discard=no' attribute, it will be kept after cardholder registration via enrolment or Activation During Shopping has completed. Otherwise, it will be removed from card data.

Note

If an encryption key is defined for the card issuer/group:

Card Number, **Name**, **PAM**, **HINT**, **HINT Response** and **Data.Value** should be sent encrypted and the Registration Server will need to decrypt these fields before using them.

Refer to section **Cardholder Registration DTD** for further details of the requirements for the encryption/decryption process.

The pre-registration data may include existing customer information such as date of birth, mother's maiden name, card verification check, credit limit, billing address, etc or a registration number distributed by mail (or in some cases a combination of both). The requirements for pre-registration information are at the discretion of member banks. The pre-registration request is dynamic enough to meet the authentication requirements of each individual member bank. However, the enrolment module determines the presentation of authentication data to the cardholder during the enrolment process.

Member banks, which require greater flexibility in the presentation or control of the enrolment process, should use the final registration model.

The pre-registration data may be removed once a cardholder has successfully completed the enrolment process.

Multiple unsuccessful enrolment attempts may cause the cardholder to be locked. This limit can be set on a per issuer basis and through the ActiveAccess administration interface.

To disable a cardholder registration, a cancel registration request should be used. To un-register finally registered cardholder, the cardholder account should be cancelled first and a new pre-registration message should be sent. This will require the cardholder to repeat the enrolment process.

A sample pre-registration request is displayed below. The following request will provide pre-registration information, which is required for enrolment of Mr. Joe Citizen for American Express SafeKey, Mastercard SecureCode and Verified by Visa.

SAMPLE PRE-REGISTRATION REQUEST

```
<?xml version="1.0"?>

<Message>

<Request Id="request1" IssuerId="1234567890123456789">

    <PreReg>
```

```
        <DataFormat Name="birthdate" Type="date" Format="YYYYMMDD" Label="Date of
        Birth:"/>

        <DataFormat Name="credit" Type="number" Label="Credit Limit:"
Desc="Please
        enter your credit card limit"/>

        <DataFormat Name="regpassword" Type="string" Label="Registration
Password:"
        Desc="Please enter your registration password" Mask="Yes"/>

        <Card Type="SPA" Number="5012345678901234" Name="Joe Citizen">

            <ClientId>819737457046382</ClientId>

            <ExpDate>202204</ExpDate>

            <Data Name="birthdate" Value="19730201"/>

            <Data Name="credit" Value="10000"/>

            <Data Name="regpassword" Value="pro2345"/>

        </Card>

        <Card Type="VbV" Number="4012345678901234" Name="Joe Citizen">

            <ClientId>819737457046382</ClientId>

            <ExpDate>202202</ExpDate>

            <Data Name="birthdate" Value="19730201"/>

            <Data Name="credit" Value="3000"/>

            <Data Name="regpassword" Value="pro2345"/>

        </Card>

        <Card Type="SK" Number="373700000000000" Name="Joe Citizen">

            <ClientId>819737457046382</ClientId>

            <ExpDate>202204</ExpDate>

            <Data Name="birthdate" Value="19730201"/>

            <Data Name="credit" Value="10000"/>

            <Data Name="regpassword" Value="pro2345"/>
```

```
        </Card>
    </PreReg>
</Request>
<Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
<!--the request signature here-->
</Signature>
</Message>
```

Final Registration Request

Members who wish to have greater control over the cardholder enrolment process typically use the final registration request. In this case the member bank itself handles the enrolment process. The registration process should result in the selection of an authentication password between the member bank and the cardholder. A personal assurance message (PAM) should also be set for 3-D Secure cards. If the issuer also supports two-factor authentication for 3-D Secure, device information will need to be set. The member bank will then format a final registration request and provide the agreed authentication information to the enrolment module for storage.

<FinalReg>		
Attributes	**Description**	**Usage**
None	N/A	N/A
Elements	**Description**	**Usage**

<FinalReg>		
<DataFormat>	Defines a data type, which can be associated with card data. This issuer-defined field has a maximum field length of 1024 characters. You can define the following attributes with the DataFormat: **Name:** The name as used by the program to refer to this data format. Also used by the data element in order to refer to this particular type, e.g.: pass or password **Label:** A short description to appear before the data elements of this type when displayed to the cardholder, e.g.: SecureCode: **Description:** A longer description to appear after the data elements of this type when displayed to the cardholder (optional), e.g.: Please enter your SecureCode **MaxLen:** The maximum length that can be stored in the data elements of this type (optional). **Type:** Can be set to date, string, number or hidden values. When set to date, number, the application performs type validation for the content of the data elements. If not specified the string type is assumed (no validation). If the data format type is set to hidden, this excludes the data format as an authentication data element. Hidden data types are not displayed to the cardholder and can be used as internal field in the XSL pages per bank to achieve certain customised features. You should not use this type unless specifically advised by GPayments to do so **Format:** Additional formatting information can be set to YYYYMMDD or YYYYMM. This is only meaningful when you have set the data format type to date. **Mask:** Can be set to Yes or No. Determines whether the user input for this data format should be masked or not. Set to Yes for password type fields. **DataMode:** This is an optional attribute. Its value can be **Identity**, **identity**, **Auth**, **auth**, **Extension**, or **extension**. Data which has Identity as DataMode will be displayed on the Registration page and may be displayed on the Forgot Password and Hint/Response pages to identify the cardholder during changing or resetting the password. Data which has Auth as DataMode will be displayed during authentication, including the Authentication and Reactivation pages. Data which has Extension as Datamode will be checked by extensions. If it is not set, the default value will be used. In the PreReg file, the default value is Identity.	Required A valid data format must be defined for each type, which is referenced by a element.

<FinalReg>		
<Card>	Card related data including card number, name on card, expiry date and issuer defined data. Card data may also include device information for two-factor authentication enabled cards. Card numbers of 19 digits are accepted and the name on card field has a maximum length of 128 characters. Device information includes, Device type {1: VASCO, 3: SMS, 7: OOB, 6: Email, 8: Decoupled Authentication} and for tokens: serial no which is a unique number assigned to each device by the manufacturer; for SMS: mobile number; and for Email: email address.	Required At least one <Card> must be present

Note

If an encryption key is defined for the card issuer/group:

Card Number, **Name**, **PAM**, **HINT**, **HINT Response**, **Data.Value** and Device.SerialNo should be sent encrypted and the Registration Server will need to decrypt these fields before using them.

Refer to section **Critical Card Data Encryption and Decryption** for further details of the requirements for the encryption/decryption process.

Cardholders registered using the final registration method can perform authenticated transactions as soon as the registration request is processed, without the need for any direct interaction with the enrolment module.

This method also allows issuers to enrol cardholders by using existing authentication data. For example, a member bank may choose to send the Internet banking username and password for authentication. Any change in username or password of the Internet banking site can also be reflected on the enrolment module by sending a new **FinalReg** request to keep the two systems synchronised.

The following request will provide final registration information, which is required for Mr. Joe Citizen to use his cards with traditional American Express SafeKey, Mastercard SecureCode and Verified by Visa.

SAMPLE FINAL REGISTRATION REQUEST FOR TRADITIONAL 3-D SECURE

```
<?xml version="1.0"?>

<Message>

<Request Id="request1" IssuerId="1234567890123456789">

    <FinalReg>

        <DataFormat Name="Password" Type="string" Label="Password:" Desc="Please
enter
        your authentication password " Mask="Yes"/>
```

```
            <DataFormat Name="Country" Type="singleSelect" Label="Please choose
the where you live in: " Desc="Please enter your country " Mask="No" MaxLen="64">

            <Option Label="Australia" Value="100"/>

            <Option Label="Germany" Value="200"/>

        </DataFormat>

            <DataFormat Name="City" Type="multiSelect" Label="Please choose the
cities you have been to: " Desc="Please select the cities " Mask="No"
MaxLen="64">

            <Option Label="Sydney" Value="11"/>

            <Option Label="Melbourne" Value="12"/>

            <Option Label="Brisbane" Value="13"/>

            <Option Label="Gold Coast" Value="14"/>

        </DataFormat>

        <Card Type="SPA" Number="5012345678901234" Name="Joe Citizen">

            <ClientId>819737457046382</ClientId>

            <ExpDate>202204</ExpDate>

            <PAM>I am sure that this is my bank</PAM>

            <Data Name="Password" Value="409634"/>

        </Card>

        <Card Type="VbV" Number="4012345678901234" Name="Joe Citizen">

            <ExpDate>202202</ExpDate>

            <PAM>I am sure that this is my bank</PAM>

            <HINT>Your childhood hero</HINT>

            <HINTResponse>Superman</HINTResponse>

            <Data name="Password" Value="354607"/>

        </Card>

        <Card Type="SK" Number="373700000000000" Name="Joe Citizen">
```

```
            <ExpDate>202204</ExpDate>
            <PAM>I am sure that this is my bank</PAM>
            <Data Name="Password" Value="409634"/>
        </Card>
        <Card Name="Joe Citizen" Number=" 3528457610673260" Type="JCB">
            <PAM>My personal message</PAM>
            <HINT>You know</HINT>
            <HINTResponse>Dreams come true</HINTResponse>
            <Data Name="Password" Value="123456">
            </Data>
            <Data Name="Country" Value="100">
            </Data>
            <Data Name="City">
                <SelectedOption Value="11"/>
                <SelectedOption Value="13"/>
            </Data>
        </Card>
    </FinalReg>
</Request>
<Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
<!--the request signature here-->
</Signature>
</Message>
```

And this request will provide final registration information, which is required for Mr. Joe Citizen to use his cards with two-factor authentication over American Express SafeKey, Mastercard SecureCode and Verified by Visa.

SAMPLE FINAL REGISTRATION REQUEST FOR TWO-FACTOR AUTHENTICATION OVER 3-D SECURE

```
<?xml version="1.0"?>

<Message>

<Request Id="request1" IssuerId="1234567890123456789">

    <FinalReg>

        <DataFormat Name="Password" Type="string" Label="Password:" Desc="Please
enter
        your authentication password " Mask="Yes"/>

        <Card Type="SPA" Number="5012345678901234" Name="Joe Citizen">

            <ClientId>819737457046382</ClientId>

            <ExpDate>202204</ExpDate>

            <PAM>I am sure that this is my bank</PAM>

            <Data Name="Password" Value="409634"/>

            <Device>

                <DeviceType>1</DeviceType>

                <SerialNo>0097123456</SerialNo>

            </Device>

        </Card>

        <Card Type="VbV" Number="4012345678901234" Name="Joe Citizen">

            <ExpDate>202202</ExpDate>

            <PAM>I am sure that this is my bank</PAM>

            <HINT>Your childhood hero</HINT>

            <HINTResponse>Superman</HINTResponse>

            <Data name="Password" Value="354607"/>
```

```
        <Device>
            <DeviceType>2</DeviceType>
            <SerialNo>79722647</SerialNo>
        </Device>
    </Card>
    <Card Type="SK" Number="3712345678901234" Name="Joe Citizen">
        <ExpDate>202204</ExpDate>
        <PAM>I am sure that this is my bank</PAM>
        <HINT>Your childhood hero</HINT>
        <HINTResponse>Superman</HINTResponse>
        <Data name="Password" Value="465809"/>
        <Device>
            <DeviceType>3</DeviceType>
            <SerialNo>+61400000000</SerialNo>
            <Param Name="SMSC">SMSGateWay</Param>
        </Device>
        <Device>
            <DeviceType>6</DeviceType>
            <SerialNo>username@domain.com</SerialNo>
        </Device>
        <Device>
            <DeviceType>7</DeviceType>
            <Param Name="OOBAdapter">sample-oob-adapter</Param>
        </Device>
        <Device>
```

```
                <DeviceType>7</DeviceType>
                <Param Name="OOBAdapter">restful-adapter</Param>
                <Param Name="OOBDeviceId">97d77002-b0b0-494e-922e-9e4cddfc6db9</
Param>
            </Device>
        </Card>
    </FinalReg>
</Request>
<Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
<!--the request signature here-->
</Signature>
</Message>
```

Cancel Registration Request

Cancel registration can be used to remove cardholder data (either pre-registration or full registration data). A card that has been cancelled can no longer be used in authenticated payments.

Cancel registration is typically used when a cardholder's account is closed or cancelled. A **CancelReg** request should consist of simple blocks of card data. The registration module ignores the content of the element, as it only requires the card number (and optionally name on card) for cancelling the cardholder's registration.

The following listing cancels the American Express SafeKey, Mastercard SecureCode and Verified by Visa registration of Mr. Joe Citizen.

> **Note**
>
> If an encryption key is defined for the card issuer/group:
>
> **Card** Number and **Name** should be sent encrypted and the Registration Server will need to decrypt these fields before using them.
>
> Refer to section **Cardholder Registration DTD** for further details of the requirements for the encryption/decryption process.

SAMPLE CANCEL REGISTRATION REQUEST

```
<?xml version="1.0"?>

<Message>

<Request Id="request1" IssuerId="123456789012345678">

    <CancelReg>

        <Card Type="SPA" Number="5012345678901234" Name="Joe Citizen"></Card>

        <Card Type="VbV" Number="4012345678901234"></Card>

        <Card Type="SK" Number="3712345678901234" Name="Joe Citizen"></Card>

    </CancelReg>

</Request>

<Signature xmlns="http://www.w3.org/2000/09/xmldsig#">

<!--the request signature here-->

</Signature>

</Message>
```

Update Registration Request

An update registration is used to update card information, such as card number, name on card, expiry date, PAM, Hint/Response and status (`Enabled` / `Disabled`). It is used to update cardholder data without affecting the cardholder's enrolment status. For example, if the cardholder's card number has changed, it is possible to save the cardholder from going through the enrolment process again by simply updating the card number.

The following listing shows how Mr. Joe Citizen's card number can be updated. Mr. Citizen will be able to continue making authenticated payments without the need to re-enrol.

SAMPLE UPDATE REGISTRATION REQUEST

```
<?xml version="1.0"?>

<Message>

<Request Id="request1" IssuerId="123456789012345678">
```

```
    <UpdateReg>
        <CardUpdate Number="5012345678901234" Name="Joe Citizen">
            <Number>5012345678943210</Number>
            <ClientId>819737457046383</ClientId>
            <ExpDate>202304</ExpDate>
            <Status>Enabled</Status>
        </CardUpdate>
        <CardUpdate Number="5123456789012353" Name="Joe Citizen">
            <ClientId>819737457046384</ClientId>
            <ExpDate>202304</ExpDate>
            <Status>Disabled</Status>
        </CardUpdate>
    </UpdateReg>
</Request>
<Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
<!--the request signature here-->
</Signature>
</Message>
```

In cases where a cardholder's account number changes, either to another card number of the same card scheme or to another card scheme, an update registration can be used to change these details. By using a CardCopy, the account details on the existing card will be transferred to a new card, with both accounts being enabled at the same time. The issuer must then disable or cancel the original account. CardCopy allows any of the card data format parameters and card attributes to be changed.

> **Note**
>
> If the Issuer is configured to use Mastercard Identity Check, the registered Mastercard cards of the issuer cannot be copied. This is because devices are not copied during CardCopy and Mastercard Identity Check requires at least one device to be assigned to the card.

> **Note**
>
> If an encryption key is defined for the card issuer/group:
>
> **Card** Number, **Name**, **PAM**, **HINT**, **HINT Response** and **Data.Value** should be sent encrypted and the Registration Server will need to decrypt these fields before using them.
>
> Refer to section **Cardholder Registration DTD** for further details of the requirements for the encryption/decryption process.

Card Device Update Request

SAMPLE CARD DEVICE UPDATE REQUEST

```
<Message>

<Request Id="request1" IssuerId="123456789012345678">

    <DeviceUpdateReg>

        <CardDeviceUpdate Number="5012345678901234" Name="Joe Citizen">

            <DeviceUpdate Operation="Add">

                <Device>

                    <DeviceType>7</DeviceType>

                    <Param Name="OOBAdapter">restful-adapter</Param>

                    <Param
Name="OOBDeviceId">123456123456123456123456123456123477</Param>

                </Device>

                <Device>

                    <DeviceType>7</DeviceType>

                    <Param Name="OOBAdapter">restful-adapter</Param>
```

```
                    <Param
Name="OOBDeviceId">777666666666666666666666666666666799</Param>
                </Device>
                <Device>
                    <DeviceType>7</DeviceType>
                    <Param Name="OOBAdapter">sample-oob-adapter</Param>
                    <Param
Name="OOBDeviceId">888666666666666666666666666666666899</Param>
                </Device>
                <Device>
                    <DeviceType>7</DeviceType>
                    <Param Name="OOBAdapter">sample-oob-adapter</Param>
                    <Param
Name="OOBDeviceId">888666666666666666666666666666666899</Param>
                </Device>
                <Device>
                    <DeviceType>7</DeviceType>
                    <Param Name="OOBAdapter">sample-oob-adapter</Param>
                </Device>
            </DeviceUpdate>
            <DeviceUpdate Operation="Delete">
                <Device>
                    <DeviceType>3</DeviceType>
                    <SerialNo>+4111112323</SerialNo>
                    <Param Name="SMSC">SMSC</Param>
                </Device>
                <Device>
```

```
                    <DeviceType>3</DeviceType>
                    <SerialNo>+641111</SerialNo>
                    <Param Name="SMSC">SMSC</Param>
                </Device>
                <Device>
                    <DeviceType>7</DeviceType>
                    <Param Name="OOBAdapter">restful-adapter1</Param>
                </Device>
                <Device>
                    <DeviceType>7</DeviceType>
                    <Param Name="OOBAdapter">restful-adapter</Param>
                    <Param
Name="OOBDeviceId">66666666666666666666666666666666777</Param>
                </Device>
                <Device>
                    <DeviceType>7</DeviceType>
                    <Param Name="OOBAdapter">restful-adapter</Param>
                    <Param
Name="OOBDeviceId">12345612345612345612345612345612345</Param>
                </Device>
            </DeviceUpdate>
        </CardDeviceUpdate>
        <CardDeviceUpdate ClientId="123456789012345">
            <DeviceUpdate Operation="Add">
                <Device>
                    <DeviceType>3</DeviceType>
                    <SerialNo>+4411111</SerialNo>
```

```
                    <Param Name="SMSC">SMSC</Param>
                </Device>
                <Device>
                    <DeviceType>7</DeviceType>
                    <Param Name="OOBAdapter">restful-adapter</Param>
                </Device>
            </DeviceUpdate>
            <DeviceUpdate Operation="Delete">
                <Device>
                    <DeviceType>3</DeviceType>
                    <SerialNo>+6411112323</SerialNo>
                </Device>
            </DeviceUpdate>
        </CardDeviceUpdate>
    </DeviceUpdateReg>
</Request>
<Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
<!--the request signature here-->
</Signature>
</Message>
```

Notification

A Notification is a record of a single cardholder event. Each event is stored in ActiveAccess and a record is logged in the following events:

- Cardholder completes their registration

- Cardholder re-activates its account by registering a new or an existing device during authentication
- Administrator registers a new or an existing device for a specified cardholder via MIA
- Administrator unregisters a device from a specified cardholder
- Administrator removes a device from a specified cardholder
- Cardholder opts-out of Activation During Shopping
- Cardholder locks their account.

At regular intervals, ActiveAccess runs a procedure to collect all notification events, storing them in separate files based on the event type. The files are stored on the server and made available for a short period. To download notifications, an issuer should send a Notification Request, requesting the notification period and the type of notification event. ActiveAccess will return the data in the form of a Notification Response message as outlined in the following sections. If the request is unable to be processed entirely, an error code, error message and error details should be sent back in the response message.

Events that can be retrieved by this method are:

- Card registration
- Card device update
- Card lock and unlock
- ADS Opt-out

XML Message Format

XML messages must be produced in accordance with the Messaging Requirements, as specified in Section **Notification DTD**.

All request and response messages are enclosed by the **<Message>** element. A message can contain either a **Request and a Signature** or a **Response**.

A message can contain either a NotificationRequest and a Signature or a NotificationResponse

```
<?xml version="1.0"?>

<Message>

<NotificationRequest Id="Notification1" IssuerId="123456789012345678"
```

```
Type="<!–Type of the query here (must be either cardreg, carddeviceupdate,
cardoptout or cardlock) -->">

<StartDate><!-- the query start date here. For example
200901010000--></StartDate>

<EndDate><!-- the query end date here. For example 200901150000
--></EndDate>

</NotificationRequest>

<Signature xmlns="http://www.w3.org/2000/09/xmldsig#">

<!--the request signature here-->

</Signature>

</Message>
```

<Message>		
Attributes	**Description**	**Usage**
None	N/A	N/A
Elements	**Description**	**Usage**
<NotificationRequest>	Used in the request message sent by the issuer to query cardholder events.	Required for request messages
<Signature>	Issuer signature is used in the request message to prove the identity of the issuer and to validate that the message content has not been altered.	Required for request message
<NotificationResponse>	The query response sent back in response to a notification request.	Required for response message

Request

A Request is sent by the issuer to query various cardholder events. A request can be sent to query Card Registration, Opt-out or Card Lock notifications.

The issuer sends a NotificationRequest requesting the period of time and the type of notification event data they require. A request may be of the type: CardReg, CardLock or CardOptOut.

<NotificationRequest>		
Attributes	**Description**	**Usage**
Id	An arbitrary identifier, which the issuer defines and can be used to refer to the request element as part of a standard URI. Also referenced by the Signature element. XML signature requires the element, which is being signed, to be identified by a unique The value entered should start with an alphabetic character.	Required
IssuerId	A unique identifier for the issuer. Created when the issuer first signs up with the system and supplied during the issuer registration process. Typically an 18 digit numeric value.	Required if GroupId not provided
GroupId	A unique identifier for the group of issuers. Created when the group is first created with the system upon system admin request. Typically an 18 digit numeric value.	Required if IssuerId not provided
Type	Determines the type of notification.	Can either be"cardreg", "carddeviceupdate", "cardlock" or "cardoptout".
EncVectorIV	If an encryption keystore has been defined for the issuer, or a group of issuers, critical card data must be encrypted using it. The critical data is encrypted using AES/CBC/PKCS5Padding mode, which requires an IV, including 16 random bytes, as an input parameter for encryption and decryption. The IV should be sent to the server to indicate that the card data in the response should be encrypted in CBC mode. To do this, the IV itself must be encrypted in AES/ECB/ PKCS5Padding mode, using an encryption key, then base64 encoded and set as EncVectorIV in the request.	Optional, if present, the server must encrypt critical card information in CBC mode, otherwise ECB or plain mode will be used instead.
Elements	**Description**	**Usage**

<NotificationRequest>		
<StartDate>	The start date of the period of interest with a format of YYYYMMDDHHMM in GMT.	Required
<EndDate>	The end date of the period of interest with a format of YYYYMMDDHHMM in GMT.	Required

Response

ActiveAccess returns the data in the form of a NotificationResponse message as outlined in the following sections. If the request is unable to be processed entirely, an error code, error message and error details are sent back in the response message.

<NotificationResponse>		
Attributes	**Description**	**Usage**
Id	The same NotificationRequest ID is returned	Required
IssuerId	Specifies the issuer ID that this report is provided for. Should be the same as the IssuerId of the request.	Required if GroupId not provided
GroupId	Specifies the group ID that this report is provided for. Should be the same as the GroupId of the request.	Required if IssuerId not provided
Type	Determines the type of response.	Can be cardreg", "carddeviceupdate", "cardlock" or "cardoptout.

<NotificationResponse>		
EncVectorIV	If the encryption keystore has been defined for the issuer or group of issuerscritical card data is encrypted using AES/ECB/PKCS5Padding or DESede/ECB/PKCS5Padding mode depending on encryption key algorithm. If the EncVectorIV attribute is set to protect the data using AES/CBC/PKCS5Padding or DESede/CBC/PKCS5Padding mode, a new random IV is generated and used by the server to encrypt critical card data in AES/CBC/PKCS5Padding or DESede/CBC/PKCS5Padding mode. When the process is finished, the IV itself is encrypted in AES/ECB/PKCS5Padding or DESede/ECB/PKCS5Padding mode using same encryption key, then base64 encoded and set as EncVectorIV in the response.	Required, if this attribute is set for the request. Server generates a new IV parameter and encrypts critical card information in the response using the CBC mode and the new IV, otherwise no attribute is set and ECB or plain mode will be used instead.
Elements	**Description**	**Usage**
<CardReg>	Contains the list of the cards, which have had registration activity within the specified period.	Required for a card registration notification request
<CardDeviceUpdate>	Contains the list of the cards, which have had device update activity, within the specified period.	Required for a card device update notification request
<CardOptOut>	Contains the list of the cards, which have opted out of their ADS within the specified period.	Required for a card opt-out notification request
<CardLock>	Contains the list of the cards, which have been locked or unlocked within the specified period.	Required for a card lock notification request
<Code>	Response code. Denotes an error has occurred in processing the request.	Required if error occurs
<ErrorMessage>	A descriptive message that identifies the category of the error.	Required if error occurs
<ErrorDetail>	A more detailed description of the error.	Required if error occurs

> **Note**
>
> All dates in notification reports will be in GMT and clients can convert them to their local time if they required.

Notifications

Card Registration Notification

The Card Registration Notification is created when the cardholder changes their registration status to confirmed or registered. A status of confirmed is attained if the issuer is using the confirmation method and the cardholder updates their default authentication password. A status of registered is achieved if the issuer chooses not to use the confirmation method and the cardholder finalises their registration through the enrolment process, Final registration or ADS. ActiveAccess should record the method of registration, time of registration and details of probable registered authentication device.

<CardReg>		
Attributes	**Description**	**Usage**
None	N/A	N/A
Elements	**Description**	**Usage**
<Card>	Card related data used to uniquely identify the cardholder including Card Number, Name on Card and Card Type.	Optional, If any related activity reported in the requested period.

> **Note**
>
> If an encryption key is defined for the card issuer/group:
>
> **Card** Number, **Card Name** and Device.**SerialId** of probable registered authentication device are encrypted in response and the client will need to decrypt these fields before using them.
>
> Refer to section **Cardholder Registration DTD** for further details of the requirements for the encryption/decryption process.

SAMPLE REGISTRATION NOTIFICATION

```
<?xml version="1.0"?>
```

```
<Message>

<NotificationResponse Type="cardreg" Id="notification1"
IssuerId="123456789012345678">

<CardReg>

<Card Type="VbV" Number="4012345678901234" Name="Joe Citizen"
ClientId="123456789012345">

<Device DeviceType="3" SerialId="+61411000001"/>

<Register>20090113080000</Register>

</Card>

</CardReg>

</Notification>

</Message>
```

Card Device Update Notification

The Card Device Update Notification is created when any of the following scenarios occur and the registered devices of the cardholder's account are changed:

- Administrator registers a new device for the specified cardholder via MIA
- Administrator registers an existing device for the specified cardholder via MIA
- Cardholder re-activates its account by registering a new device during authentication
- Cardholder re-activates its account by registering an existing device during authentication
- Administrator unregisters a device of the specified cardholder
- Administrator removes a device of the specified cardholder

<CardDeviceUpdate>		
Attributes	**Description**	**Usage**
None	N/A	N/A
Elements	**Description**	**Usage**

<CardDeviceUpdate>		
<Card>	Card related data used to uniquely identify the cardholder including Card Number, Name on Card and Card Type.	Optional, If any related activity reported in the requested period.

Note

If an encryption key is defined for the card issuer/group:

Card Number, **Card Name** and **DeviceUpdate.SerialId** of the device involved, are encrypted in the response and will need to be decrypted before they can be used.

Refer to section **Cardholder Registration DTD** for further details of the requirements for the encryption/decryption process.

SAMPLE DEVICE UPDATE NOTIFICATION

```
<?xml version="1.0" encoding="UTF-8"?>

<Message>

<NotificationResponse Id="Notification1" IssuerId="123456789012345678"
Type="carddeviceupdate">

<CardDeviceUpdate>

<Card Name="CARD19" Number="4123455000000000009" Type="VbV"
ClientId="123456789012345">

<DeviceUpdate Action="RemoveDevice" Date="20140214052643" DeviceType="SMS"
SerialId="+355665544332211"/>

<DeviceUpdate Action="UnregisterDevice" Date="20140214044825" DeviceType="SMS"
SerialId="+355112233445566"/>

<DeviceUpdate Action="RegisterExistingDevice" Date="20140214044255"
DeviceType="SMS" SerialId="+355665544332211"/>

<DeviceUpdate Action="RegisterNewDevice" Date="20140214044202" DeviceType="SMS"
SerialId="+355112233445566"/>

</Card>

</CardDeviceUpdate>

</NotificationResponse>
```

```
</Message>
```

Card Opt-Out Notification

The Card Opt-Out Notification is created when the cardholder opts-out of the ADS. ActiveAccess should record the date and time of each opt-out. The response should only list the date/time of the most recent opt-out and the sequence number of this opt-out within the requested period and should not list any previous opt-outs which may have occurred during the period.

<CardOptOut>		
Attributes	**Description**	**Usage**
None	N/A	N/A
Elements	**Description**	**Usage**
<Card>	Card related data used to uniquely identify the cardholder including Card Number, Name on Card and Card Type.	Optional, If any related activity reported in the requested period.

Note

If an encryption key is defined for the card issuer/group:

Card Number and **Name** are encrypted in response and the client will need to decrypt these fields before using them.

Refer to section **Cardholder Registration DTD** for further details of the requirements for the encryption/decryption process.

SAMPLE OPT-OUT NOTIFICATION

```
<?xml version="1.0"?>

<Message>

<NotificationResponse Type="cardoptout" Id="notification2"
IssuerId="123456789012345678">

<CardOptOut>

<Card Type="VbV" Number="4012345678901234" Name="Joe Citizen"
```

```
ClientId="123456789012345">
<Optout Number="2">20090131083025</Optout>
</Card>
</CardOptOut>
</Notification>
</Message>
```

Card Lock Notification

The Card Lock Notification is created when the cardholder locks their account during the registration or authentication process. ActiveAccess should record the date and time when a cardholder locks their account and where it is locked (enrolment website or ACS). This should be made available for querying, as with the other notifications.

<CardLock>		
Attributes	**Description**	**Usage**
None	N/A	N/A
Elements	**Description**	**Usage**
<Card>	Card related data used to uniquely identify the cardholder including Card Number, Name on Card and Card Type.	Optional, If any related activity reported in the requested period.

Note

If an encryption key is defined for the card issuer/group:

Card Number and **Name** are encrypted in response and the client will need to decrypt these fields before using them.

Refer to section **Cardholder Registration DTD** for further details of the requirements for the encryption/decryption process.

SAMPLE LOCK NOTIFICATION

```
<?xml version="1.0"?>
```

```
<Message>

<NotificationResponse Type="cardlock" Id="notification1"
IssuerId="123456789012345678">

<CardLock>

<Card Type="VbV" Number="4012345678901234" Name="Joe Citizen"
ClientId="123456789012345">

<Lock>20090114093015</Lock>

<Unlock>20090114095015</Unlock>

</Card>

</CardLock>

</Notification>

</Message>
```

User Registration

The authentication system is also responsible for authentication of users during any transactions, commercial or otherwise using two-factor authentication. As a result, it is necessary that the system stores user related information to the extent that this requirement can be satisfied.

There are three user registration models for populating data into the authentication system. Direct entry allows help desk operators to manually enter user registration data through the ActiveAccess administration interface. The final registration and pre-registration models both format user data in an XML message. These messages are either sent directly to the registration server via an API or uploaded through the ActiveAccess administration interface. To facilitate these processes and integrate with the authentication system some integration work with the issuer's systems is required.

Info

See the document, **ActiveAccess Member Bank Administration**, for more information on the Direct Entry user registration model.

The purpose of registration is to upload or supply the information necessary to enable a user to use their two-factor authentication device to gain access to a secured resource such as the issuer's Internet site. For each registered user, the registration message must include a username and the authentication device information.

There are four types of XML messages accepted by the system for the purpose of user creation and maintenance. They are:

- **PreReg**: enables the creation of pre-registered user data in the system
- **FinalReg**: enables the creation of fully registered user data in the system
- **UpdateReg**: enables the alteration of user data in the system
- **CancelReg**: enables the removal of user data from the system

> **Info**
>
> **Section Request** provides more detail on the format of these messages.

To perform authenticated transactions, it is essential that the user data elements stored within the authentication system are sufficient to allow this. The essential elements required are:

- Username
- Device

The information required for user authentication is primarily provided to the authentication system by the member bank. The Registration API provides a flexible data transport and definition mechanism that allows each member bank to build an individual user registration process to meet their specific requirements.

Alternatively, user data can be formatted into XML files and uploaded through the ActiveAccess administration interface. This process reduces the technical requirements for the issuer.

Pre-registration

When using the pre-registration model, a member bank only needs to establish the authentication criteria for enrolment of users and provide this information to the authentication system. The actual enrolment process is then handled by the authentication system's activation during authentication. The authentication system uses the pre-registration information to verify the identity of the user and set-up the user account during the user enrolment process.

In the pre-registration model, it is essential that a **PreReg** message uploads Username. It is also required that the issuer supplies a number of other known parameters to the authentication system to verify the user when they come to enrol with the authentication system. The following provides a list of suggested data that can be included in the **PreReg** message to allow the user authentication to occur. Please note that these fields are at the discretion of the issuing bank:

- Date of birth
- Mother's maiden name
- Internet Banking username
- Information about accounts held by the user (e.g. account types, credit limit, billing address, etc)

An example of pre-registration is when the member bank mails invitations to users and provides them with, say, a registration number. Users can then enrol using the activation during shopping process. Having entered their registration number and once the user's identity is successfully verified, they can proceed with setting up their account, which includes selection of device specification. Once the enrolment process is complete, the tokens generated by the enrolled authentication device will then be used in all subsequent authenticated transactions.

Final Registration

The final registration model allows issuers to control the user's enrolment process. It requires the member bank to develop their own enrolment process to collect user enrolment information. Once the user enrolment process is complete, the information is sent to the authentication system in the form of a final registration message.

In the final registration model, it is essential that a **FinalReg** message uploads the Username. It is also required that the issuer supplies the specifications of device which is used at authentication time for generating tokens:

- Device Type
- Device Serial No

An example of final registration is enrolment of users through the issuer's Internet banking site. A possible scenario is for the user to log into the Internet banking site and enable two-factor authentication for their account. The user will then be prompted to choose device type and its serial number for their account. The issuer then formats a final registration message, which is sent to the authentication system in order to complete the user enrolment.

The format of the registration request is the same for real-time and batch uploads, except that the batch upload may contain multiple blocks of user data.

The registration API uses XML as the message format and HTTP as the message transport. API calls (requests) can be made to the registration server of the authentication system. Issuers can use the registration API to automate their user registration process.

XML Message Format

XML messages must be produced in accordance with the Messaging Requirements, as specified in section User Registration DTD .

All request and response messages are enclosed by the <Message> element. A message can contain either a Request and a Signature or a Response.

A message can contain either a Request and a Signature or a Response

```
<?xml version="1.0"?>

<Message>

<Request Id="request1" IssuerId="1234567890123456789">

<!--the request content here-->

</request>

<Signature xmlns="http://www.w3.org/2000/09/xmldsig#">

<!--the request signature here-->

</Signature>

</Message>
```

<Message>		
Attributes	**Description**	**Usage**
None	N/A	N/A
Elements	**Description**	**Usage**

<Message>		
<Request>	Used in the request message, which is sent by the issuer to register one or more users.	Required for request messages
<Signature>	Issuer signature is used in the request message to prove the identity of the issuer and to validate that the message content has not been altered.	Required for request message
<Response>	User registration response sent back in response to a user registration request.	Required for response message

Note

The XML response starts with the XML declaration (<?xml version="1.0" encoding="UTF-8"?>). However a request does not need to start with the XML declaration. Request content must be sent with UTF-8 encoding.

Request

A Request is sent by the issuer to perform various user registration tasks. A request can be sent to perform pre-registration, final registration, cancel registration or update registration for one or more users. A request may only contain one <PreReg>, <FinalReg>, <CancelReg> or <UpdateReg>.

<Request>		
Attributes	**Description**	**Usage**
Id	An arbitrary identifier, which the issuer defines and can be used to refer to the request element as part of a standard URI. Also referenced by the Signature element. XML signature requires the element, which is being signed, to be identified by a unique The value entered should start with an alphabetic character.	Required
IssuerId	A unique identifier for the issuer. Created when the issuer first signs up with the system and supplied during the issuer registration process. Typically an 18 digit numeric value.	Required
Elements	**Description**	**Usage**

<Request>		
<PreReg>	Used for pre-registration of one or more users. Pre-registered Users will need to go through the enrolment process to finalise their registration.	Required for a pre-registration request
<FinalReg>	Used for registration of one or more users. Registered Users can start making authenticated transactions without the need to go through the enrolment process.	Required for a final registration request
<CancelReg>	Used to remove one or more users from the system.	Required for a cancel registration request
<UpdateReg>	Used to update the information for one or more users. For example to change the Username. User's registration status is left unchanged.	Required for an update registration request

Response

A response message is sent back for each request. The response message provides the result of the request message with details of errors, if any. Issuers must process the response message and should correct and replay their request if there is an error.

<Response>		
Attributes	**Description**	**Usage**
None	N/A	N/A
Elements	**Description**	**Usage**
<Code>	Response code. 0 if the request was successful. 1 if the request has been successfully processed but there are warnings. Any other value denotes an error in processing the request.	Required
<ErrorMessage>	A descriptive message that identifies the category of the error.	Required
<ErrorDetail>	A more detailed description of the error.	Required

<Response>		
<Warning>	A warning is issued to provide information on an unexpected situation that does not prevent the request from being successfully processed.	Conditional. Required only if response code is 1.

Note

A message with response code **1** denotes that the request has been successfully processed but yet the registration server has not been able to comply with certain instructions. For example a cancel registration attempt on an already cancelled user is not processed and as such a warning message is reported. An issuer does not need to take any further action in this case. Warning messages may be logged at the issuer's end for later reference.

Note

Multiple warning messages may be included in a single response message.

Requests

Pre-registration Request

The pre-registration request is typically used by the members who wish to leave user enrolment to authentication time. A pre-registered user cannot perform authenticated transactions until they finalise the registration by going through the standard enrolment process, which is offered by the user's activation during authentication process. Pre-registration data is used to verify the identity of users during the enrolment process.

<PreReg>		
Attributes	**Description**	**Usage**
None	N/A	N/A
Elements	**Description**	**Usage**

<PreReg>		
<DataFormat>	Defines a data type, which can be associated with user data. This issuer-defined field has a maximum field length of 1024 characters. You can define the following attributes with the DataFormat: **Name:** The name as used by the program to refer to this data format. Also used by the data element in order to refer to this particular type, e.g.: pass or password **Label:** A short description to appear before the data elements of this type when displayed to the cardholder, e.g.: Birth Date: **Description:** A longer description to appear after the data elements of this type when displayed to the cardholder (optional), e.g.: Please enter your birth date **MaxLen:** The maximum length that can be stored in the data elements of this type (optional). **Type:** Can be set to date, string, number or hidden values. When set to date, number, the application performs type validation for the content of the data elements. If not specified the string type is assumed (no validation). If the data format type is set to hidden, this excludes the data format as an authentication data element. Hidden data types are not displayed to the users and can be used as internal field in the XSL pages per bank to achieve certain customised features. You should not use this type unless specifically advised by GPayments to do so **Format:** Additional formatting information can be set to YYYYDDMM or YYYYMM. This is only meaningful when you have set the data format type to date. **Mask:** Can be set to Yes or No. Determines whether the user input for this data format should be masked or not. Set to Yes for password type fields.	Required A valid data format must be defined for each type, which is referenced by a <Data> element.
<UserReg>	User related data including username, name, password and issuer defined data. Usernames up to 128 characters are accepted and the name field has a maximum length of 256 characters.	Required At least one <UserReg> must be present

The pre-registration data may include existing customer information such as date of birth, mother's maiden name, card verification check, credit limit, billing address, etc or a registration number distributed by mail (or in some cases a combination of both). The requirements for pre-registration information are at the discretion of member banks. The pre-registration request is dynamic enough to meet the authentication requirements of each individual member bank.

Member banks, which require greater flexibility in the presentation or control of the enrolment process, should use the final registration model.

The pre-registration data may be removed once a user has successfully completed the enrolment process.

Multiple unsuccessful enrolment attempts may cause the user to be locked. This limit can be set on a per issuer basis and through the ActiveAccess administration interface.

To disable a user registration, a cancel registration request should be used. To un-register finally registered users, the user account should be cancelled first and a new pre-registration message should be sent. This will require the user to repeat the enrolment process.

A sample pre-registration request is displayed below. The following request will provide pre-registration information, which is required for enrolment of Mr. Joe Citizen:

SAMPLE PRE-REGISTRATION REQUEST

```
<?xml version="1.0"?>

<Message>

<Request Id="request1" IssuerId="1234567890123456">

<PreReg>

<DataFormat Name="birthdate" Type="date" Format="YYYYMMDD" Label="Date of
Birth:"/>

<DataFormat Name="regpassword" Type="string" Label="Registration Password:"
Desc="Please enter your registration password" Mask="Yes"/>

<UserReg Username="citizenjoe">

<Name>Mr. Joe Citizen</Name>

<Password>123456</Password>

<Data Name="birthdate" Value="19730201"/>

<Data Name="regpassword" Value="pro2345"/>

</UserReg>

</PreReg>

</Request>

<Signature xmlns="http://www.w3.org/2000/09/xmldsig#">

<!--the request signature here-->
```

```
</Signature>

</Message>
```

Final Registration Request

Members who wish to have greater control over the user enrolment process typically use the final registration request. In this case the member bank itself handles the enrolment process. The registration process should result in the setup of a device between the member bank and the user. The member bank will then format a final registration request and provide the agreed authentication information to the enrolment module for storage.

<FinalReg>		
Attributes	**Description**	**Usage**
None	N/A	N/A
Elements	**Description**	**Usage**
<UserReg>	User related data including username, name, an optional password and device information. Usernames with maximum 128 characters and names that has a maximum length of 256 characters are accepted. Device information includes, Device type {1: VASCO, 3: SMS, 7: OOB, 6: Email, 8: Decoupled Authentication} specifies type of device which will be used for generating tokens and serial no which is a unique identifier assigned by the device manufacturer.	Required At least one <UserReg> must be present

Users registered using the final registration method can perform authenticated transactions as soon as the registration request is processed, without the need to go through activation during authentication process.

Note that the 'Password' field is optional and is only used when both the first and the second factor of authentication is to be handled by ActiveAccess. Typically an issuer would use their existing first factor authentication (e.g. Internet banking password) to authenticate the user themselves and would only rely on ActiveAccess for second factor authentication.

The following request will provide final registration information, which is required for Mr. Joe Citizen.

SAMPLE FINAL REGISTRATION REQUEST

```
<?xml version="1.0"?>
<Message>
<Request Id="request1" IssuerId="1234567890123456789">
<FinalReg>
<UserReg Username="citizenjoe">
<Name>Mr. Joe Citizen</Name>
<Password>123456</Password>
<Device>
<DeviceType>1</DeviceType>
<SerialNo>0097123456</SerialNo>
</Device>
</UserReg>
</FinalReg>
</Request>
<Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
<!--the request signature here-->
</Signature>
</Message>
```

Cancel Registration Request

Cancel registration can be used to remove user data (either pre-registration or full registration data). A user that has been cancelled can no longer be used in authentication process.

Cancel registration is typically used when a user's account is closed or cancelled. A **CancelReg** request should consist of simple blocks of user data. The enrolment module ignores the content of the <User> element, as it only requires the Username for cancelling the user's registration.

The following listing cancels the registration of Mr. Joe Citizen.

SAMPLE CANCEL REGISTRATION REQUEST

```
<?xml version="1.0"?>

<Message>

<Request Id="request1" IssuerId="1234567890123456789">

<CancelReg>

<User Username="citizenjoe"></User>

</CancelReg>

</Request>

<Signature xmlns="http://www.w3.org/2000/09/xmldsig#">

<!--the request signature here-->

</Signature>

</Message>
```

Update Registration Request

An update registration is used to update user information such as username, name and password. It is used to update user data without affecting the user's enrolment status. For example if the user's username has changed it is possible to save the user from going through the enrolment process again by simply updating the username.

To update user authentication data, use pre-registration or final registration request.

The following listing shows how Mr. Joe Citizen's username and name can be updated. Mr. Citizen will be able to continue making authenticated logins without the need to re-enrol.

SAMPLE UPDATE REGISTRATION REQUEST

```
<?xml version="1.0"?>

<Message>

<Request Id="request1" IssuerId="1234567890123456789">

<UpdateReg>
```

```
<UserUpdate Username="Joe Citizen">
<Name>Mr. Ko Citizen</Name>
<Username>citizenco</Username>
<Password>123456</Password>
</UserUpdate>
</UpdateReg>
</Request>
<Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
<!--the request signature here-->
</Signature>
</Message>
```

Messaging Requirements

Signing the Message

A request must be always accompanied by a Signature element to determine the authenticity of the message. Issuers are required to sign the <Request>….</Request> element of their message. As the registration server is not a general XML signature verifier tool, it does not resolve the specified URI in <Reference> of the signed information and always assumes that the client has signed the <Request>….</Request> element.

The following table summarises the algorithms that should be used for signing the XML message and their reference.

Algorithm	Reference
Canonicalization	http://www.w3.org/TR/2001/REC-xml-c14n-20010315
Message Digest	http://www.w3.org/2000/09/xmldsig#sha1
Encoding	http://www.w3.org/2000/09/xmldsig#base64

Algorithm	Reference
Signature	http://www.w3.org/2000/09/xmldsig#rsa-sha1
MAC	http://www.w3.org/2000/09/xmldsig#hmac-sha1

Info

Please refer to XML-Signature Syntax and Processing, W3C Proposed Recommendation, dated 20 August 2001 at http://www.w3.org/TR/2001/PR-xmldsig-core-20010820 for further information.

Example of a message with standard XML signature

```
<?xml version="1.0"?>

<Message>

<Request Id="request1" IssuerId="123456789012345678">

<!--the request content here-->

</Request>

<Signature xmlns="http://www.w3.org/2000/09/xmldsig#">

<SignedInfo>

<CanonicalizationMethod
Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315">

</CanonicalizationMethod>

<SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1">

</SignatureMethod>

<Reference URI="#request1">

<DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1">

</DigestMethod>

<DigestValue>pHfyjnLJ2LK0Vc4cLgYSFp8gGhM=</DigestValue>

</Reference>
```

```
</SignedInfo>

<SignatureValue>

eHNXORO0egBEFqYt16zOtXG4FaraIEfCxM5cZ2QYCCl3tqbx9ynF6DmOdlLymaR0kBdkIAWS6uYC3Tg3Z8t
9i+ze4veCZLfHsXbJsvHxcAsF/kRJWmDCfpSrApbKqIkmAPWpw3W8hxF950gggYWkXOCSUIw4C1Vc=

</SignatureValue>

<KeyInfo>

<X509Data>

<X509Certificate>

MIIDwjCCA2ygAwIBAgIIGtU11QAAAGEwDQYJKoZIhvcNAQEEBQAwfjELMAkGA1UEBhMCQVUxGDAWBgNVBAg
V0aCBXYWxlczEPMA0GA1UEBxMGU3lkbmV5MRowGAYDVQQKExFHUGF5bWVudHMgUHR5IEx0ZDEUMBIGA1UEC
dmljZXMxEjAQBgNVBAMTCUdQYXltZW50czAeFw0wMjEyMDUyMzU5MTJaFw0wNjExMjkwNTM1NDRaMHYxCzA
FVMRgwFgYDVQQIEw9OZXcgU291dGggV2FsZXMxDzANBgNVBAcTBlN5ZG5leTESMBAGA1UEChMJR1BheW1lb
VQQLEwtJVCBTZXJ2aWNlczESMBAGA1UEAxMJMTI3LjAuMC4xMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQK
+EfXoR
8yljNopWKm/nnEqyBkghJc9xuO+uVkYqPTuuIK6KFVxHGU+BT3+SAtP2K5MQIUCpi9c6/
yc6wrYfFvHyW9nf6LixAsAeZ2
DiMCZfD1TUwoA0FOjwEcBollj4SkqOnZia9kuHVpkhLiOxxHJMgXuIymyfWDChikH+/
4LwIDAQABo4IBkDCCAYwwgbkGA1
UdIwSBsTCBroAUqG0JqSygT3QCMlmFHp+x41MjfzChgYOkgYAwfjELMAkGA1UEBhMCQVUxGDAWBgNVBAgTD
aCBXYWxlczEPMA0GA1UEBxMGU3lkbmV5MRowGAYDVQQKExFHUGF5bWVudHMgUHR5IEx0ZDEUMBIGA1UECxM
ljZXMxEjAQBgNVBAMTCUdQYXltZW50c4IQazV34VzY/rxAdU/
vkRbzrjB5BgNVHR8EcjBwMDWgM6Axhi9odHRwOi8vVklT
QURJUi9DZXJ0U3J2L0NlcnRFbnJvbGwvR1BheW1lbnRzLmNybDA3oDWgM4YxZmlsZTovL1xcVklTQURJUlx
NlcnRFbnJvbGxcR1BheW1lbnRzLmNybDBTBggrBgEFBQcBAQRHMEUwQwYIKwYBBQUHMAKGN2h0dHA6Ly9WS
cnRTcnYvQ2VydEVucm9sbC9WSVNBRElSX0dQYXltZW50cy5jcnQwDQYJKoZIhvcNAQEEBQADQQCrPtHfPVp
5
EXYiwDxJg1HSDrxi7Kgvf0jQD18uz6m48BQ2Pb/wQX/eMkXcQlOIAit/K7tHD8A4wG

</X509Certificate>

</X509Data>

</KeyInfo>

</Signature>

</Message>
```

The registration API ignores the content of <X509Certificate> and does not use this element for verification of the <SignatureValue>. The DTD requires at least one of the many <X509Data> elements such as <X509Certificate> to be specified however, you may leave the content of <X509Certificate> empty

(<X509Certificate> </X509Certificate>).

XML Signing Certificate

The verification of the message signature is based on the XML signing certificate. The member bank submits this certificate as part of the banks testing and enrolment process. After it is received, it is stored in the issuer's profile in the authentication system database. This is a two-party scenario whereby the authentication system operator verifies the identity of the member bank before accepting the certificate. As such there is no need for the XML signing certificate to be signed by a third party and member banks can submit a self-signed certificate.

Critical Card Data Encryption and Decryption

The key, which is used for encrypting/decrypting the critical card data, must be a 128 bit AES key. A KeyStore with the following details should be prepared for the encryption key that is to be uploaded, through MIA, for the specified issuer or group of issuers:

KeyStore type/format: JCEKS

KeyStore provider: SunJCE

Key algorithm: AES

Key size: 128 bit

Key name: can be any

No of keys in the KeyStore: Only one key must be populated in the KeyStore

Such KeyStores can be easily created by the Java keytool utility using the following command:

```
keytool -genseckey -alias enckey128 -keypass 123456 -keyalg AES -keysize 128
-keystore enc-key.JKS -storepass 123456 -storetype JCEKS
```

In order to facilitate a smooth transition to the latest version of CardLoader, keys with the following specifications are also supported:

KeyStore type/format: JCEKS

KeyStore provider: SunJCE

Key algorithm: DESede

Key size: 112 or 168 bit

Key name: can be any

If EncVectorIV is set for the request/response, the registration server/client needs to get the IV by base64 decoding and decrypting the EncVectorIV using the encryption key in DESede/ECB/PKCS5Padding or AES/ECB/PKCS5Padding mode, before decrypting the critical card data in DESede/CBC/PKCS5Padding or AES/CBC/PKCS5Padding mode using the obtained EncVectorIV from the request/response.

Cardholder Registration

In cardholder registration messages, critical cardholder information must be encrypted using AES/ECB/PKCS5Padding or AES/CBC/PKCS5Padding mode (if EncVectorIV is used for the request) in ActiveAccess v8+, and using DESede/ECB/PKCS5Padding or DESede/CBC/PKCS5Padding mode (if EncVectorIV is used for the request), and critical cardholder information must be encrypted in older versions of ActiveAccess. The output must then be base64 encoded and included in the message.

If a card encounters a warning during the process, its **Card Number**, **Name** and **Device.SerialNo** (if final registration request contains device information) would be encrypted in the response. In this case, if EncVectorIV has been set for the request, the server generates a new IV, encrypts it using DESede/ECB/PKCS5Padding or AES/ECB/PKCS5Padding mode, base64 encodes it and then sets it in the response. If the server sets EncVectorIV for the response, the IV must be obtained by base64 decoding and decrypting the EncVectorIV using the encryption key in DESede/ECB/PKCS5Padding or AES/ECB/PKCS5Padding mode before decrypting the critical card data in the response. Critical cardholder information must then be decrypted using DESede/ECB/PKCS5Padding or DESede/CBC/PKCS5Padding and AES/ECB/PKCS5Padding or AES/CBC/PKCS5Padding (If EncVectorIV has been set for the response) mode, depending on encryption key algorithm.

Notification

In notification responses, critical cardholder information must be encrypted using DESede/ECB/PKCS5Padding or DESede/CBC/PKCS5Padding and AES/ECB/PKCS5Padding or AES/CBC/PKCS5Padding (If EncVectorIV is used for the request) mode, depending on encryption key algorithm. The output must then be base64 encoded and included in the message.

Card Number, **Name**, **Device.SerialId** and **DeviceUpdate.SerialId** would be encrypted in the response. In this case, if EncVectorIV has been set for the request, the server generates a new IV,

encrypts it using DESede/ECB/PKCS5 Padding or AES/ECB/PKCS5Padding mode, base64 encodes it and then sets it in the response. If the server sets EncVectorIV for the response, the IV must be obtained by base64 decoding and decrypting the EncVectorIV, using the encryption key in DESede/ECB/PKCS5 Padding or AES/ECB/PKCS5Padding mode, before decrypting the critical card data in the response. Critical cardholder information must then be decrypted using DESede/ECB/PKCS5 Padding or DESede/CBC/PKCS5Padding and AES/ECB/PKCS5Padding or AES/CBC/PKCS5Padding (If EncVectorIV has been set for the response) mode, depending on encryption key algorithm.

Calling Convention

Requests must be sent using HTTPS POST. The HTTP header must have a Content-Length, which should be set to the body length of the request.

Registration API DTD

Issuers must make sure that their XML request conforms to the requirements of the registration API by validating the request against the appropriate DTD.

Cardholder Registration DTD

Validate the request against the following DTD.

```
<?xml version="1.0" encoding="UTF-8"?>

<!ELEMENT Message ((Request, Signature) | Response | (GetResponse, Signature) |
GetProgress | Progress)>
<!ATTLIST Message
    Id CDATA #IMPLIED
>
<!ELEMENT Request (PreReg | FinalReg | CancelReg | UpdateReg | DeviceUpdateReg)>
<!ATTLIST Request
    IssuerId NMTOKEN #IMPLIED
    GroupId NMTOKEN #IMPLIED
    Id ID #REQUIRED
    EncVectorIV CDATA #IMPLIED
>
<!ATTLIST GetResponse
    Id ID #REQUIRED
    EncVectorIV CDATA #IMPLIED
>
<!ATTLIST Response
```

```
    EncVectorIV CDATA #IMPLIED
>
<!ELEMENT Response (Code, ErrorMessage, ErrorDetail, Warning*)>
<!ELEMENT GetResponse (IssuerId?, GroupId?)>
<!ELEMENT GetProgress (IssuerId?, GroupId?)>
<!ELEMENT PreReg (DataFormat*, Card+)>
<!ELEMENT FinalReg (DataFormat*, Card+)>
<!ELEMENT CancelReg (Card+)>
<!ELEMENT UpdateReg (CardUpdate*,CardCopy*)>
<!ELEMENT Card (ClientId*, ExpDate?, PAM?, HINT?, HINTResponse?, Data*, Device*)>
<!ELEMENT Device (DeviceType, SerialNo?, Param*)>
<!ATTLIST Card
    Type (SPA | VbV | JCB | SK | DC) #REQUIRED
    Number CDATA #REQUIRED
    Name CDATA #IMPLIED
>
<!ELEMENT CardUpdate (Number?, Name?, ClientId*, ExpDate?, PAM?, HINT?,
HINTResponse?, Status?, Data*)>
<!ATTLIST CardUpdate
    Number CDATA #REQUIRED
    Name CDATA #REQUIRED
>
<!ELEMENT CardCopy (Type?, Number?, Name?, ClientId*, ExpDate?, PAM?, HINT?,
HINTResponse?, Status?, Data*)>
<!ATTLIST CardCopy
    Number CDATA #REQUIRED
    Name CDATA #REQUIRED
>
<!ELEMENT DeviceUpdateReg (CardDeviceUpdate+)>
<!ELEMENT CardDeviceUpdate (DeviceUpdate+)>
<!ATTLIST CardDeviceUpdate
    Number CDATA #IMPLIED
    Name CDATA #IMPLIED
    ClientId CDATA #IMPLIED
>
<!ELEMENT DeviceUpdate (Device+)>
<!ATTLIST DeviceUpdate
    Operation (Add | Delete) #REQUIRED
>
<!ATTLIST Param
    Name NMTOKEN #REQUIRED
>
<!ELEMENT IssuerId (#PCDATA)>
<!ELEMENT GroupId (#PCDATA)>
<!ELEMENT Progress (#PCDATA)>
<!ELEMENT Code (#PCDATA)>
<!ELEMENT ErrorMessage (#PCDATA)>
<!ELEMENT ErrorDetail (#PCDATA)>
<!ELEMENT Warning (#PCDATA)>
<!ELEMENT Type (#PCDATA)>
<!ELEMENT ExpDate (#PCDATA)>
```

```
<!ELEMENT ClientId (#PCDATA)>
<!ELEMENT Name (#PCDATA)>
<!ELEMENT Number (#PCDATA)>
<!ELEMENT PAM (#PCDATA)>
<!ELEMENT HINT (#PCDATA)>
<!ELEMENT HINTResponse (#PCDATA)>
<!ELEMENT DeviceType (#PCDATA)>
<!ELEMENT Status (#PCDATA)>
<!ELEMENT SerialNo (#PCDATA)>
<!ELEMENT Param (#PCDATA)>
<!ELEMENT SelectedOption EMPTY>
<!ATTLIST SelectedOption
    Value NMTOKEN #REQUIRED
>
<!ELEMENT Data (SelectedOption*)>
<!ATTLIST Data
    Name NMTOKEN #REQUIRED
    Value CDATA #IMPLIED
    Discard (Yes | yes | No | no) #IMPLIED
>
<!ELEMENT Option EMPTY>
<!ATTLIST Option
    Label CDATA #REQUIRED
    Value NMTOKEN #REQUIRED
>
<!ELEMENT DataFormat (Option*)>
<!ATTLIST DataFormat
    Name NMTOKEN #REQUIRED
    Label CDATA #REQUIRED
    Desc CDATA #IMPLIED
    MaxLen NMTOKEN #IMPLIED
    Type (date | string | number | hidden | singleSelect | multiSelect) #IMPLIED
    Format (YYYYMMDD | YYYYMM) #IMPLIED
    Mask (Yes | yes | No | no) #IMPLIED
    DataMode (Auth | Identity | Extension | auth | identity | extension) #IMPLIED
>
<!-- DTD for XML Signatures http://www.w3.org/2000/09/xmldsig# -->
<!ENTITY % Object.ANY ''>
<!ENTITY % Method.ANY ''>
<!ENTITY % Transform.ANY ''>
<!ENTITY % SignatureProperty.ANY ''>
<!ENTITY % KeyInfo.ANY ''>
<!ENTITY % KeyValue.ANY ''>
<!ENTITY % PGPData.ANY ''>
<!ENTITY % X509Data.ANY ''>
<!ENTITY % SPKIData.ANY ''>
<!-- Start Core Signature declarations, these should NOT be altered -->
<!ELEMENT Signature (SignedInfo, SignatureValue, KeyInfo?, Object*)>
<!ATTLIST Signature
    xmlns CDATA #FIXED "http://www.w3.org/2000/09/xmldsig#"
    Id ID #IMPLIED
```

```
>
<!ELEMENT SignatureValue (#PCDATA)>
<!ATTLIST SignatureValue
    Id ID #IMPLIED
>
<!ELEMENT SignedInfo (CanonicalizationMethod, SignatureMethod, Reference+)>
<!ATTLIST SignedInfo
    Id ID #IMPLIED
>
<!ELEMENT CanonicalizationMethod (#PCDATA %Method.ANY;)* >
<!ATTLIST CanonicalizationMethod
    Algorithm CDATA #REQUIRED
>
<!ELEMENT SignatureMethod (#PCDATA|HMACOutputLength %Method.ANY;)* >
<!ATTLIST SignatureMethod
    Algorithm CDATA #REQUIRED
>
<!ELEMENT Reference (Transforms?, DigestMethod, DigestValue)>
<!ATTLIST Reference
    Id ID #IMPLIED
    URI CDATA #IMPLIED
    Type CDATA #IMPLIED
>
<!ELEMENT Transforms (Transform+)>
<!ELEMENT Transform (#PCDATA|XPath %Transform.ANY;)* >
<!ATTLIST Transform
    Algorithm CDATA #REQUIRED
>
<!ELEMENT XPath (#PCDATA)>
<!ELEMENT DigestMethod (#PCDATA %Method.ANY;)* >
<!ATTLIST DigestMethod
    Algorithm CDATA #REQUIRED
>
<!ELEMENT DigestValue (#PCDATA)>
<!ELEMENT KeyInfo (#PCDATA|KeyName|KeyValue|RetrievalMethod|
X509Data|PGPData|SPKIData|MgmtData %KeyInfo.ANY;)* >
<!ATTLIST KeyInfo
    Id ID #IMPLIED
>
<!-- Key Information -->
<!ELEMENT KeyName (#PCDATA)>
<!ELEMENT KeyValue (#PCDATA|DSAKeyValue|RSAKeyValue %KeyValue.ANY;)* >
<!ELEMENT MgmtData (#PCDATA)>
<!ELEMENT RetrievalMethod (Transforms?)>
<!ATTLIST RetrievalMethod
    URI CDATA #REQUIRED
    Type CDATA #IMPLIED
>
<!-- X.509 Data -->
<!ELEMENT X509Data ((X509IssuerSerial | X509SKI | X509SubjectName |
X509Certificate | X509CRL )+ %X509Data.ANY;)>
```

```
<!ELEMENT X509IssuerSerial (X509IssuerName, X509SerialNumber)>
<!ELEMENT X509IssuerName (#PCDATA)>
<!ELEMENT X509SubjectName (#PCDATA)>
<!ELEMENT X509SerialNumber (#PCDATA)>
<!ELEMENT X509SKI (#PCDATA)>
<!ELEMENT X509Certificate (#PCDATA)>
<!ELEMENT X509CRL (#PCDATA)>
<!-- PGPData -->
<!ELEMENT PGPData ((PGPKeyID, PGPKeyPacket?) | (PGPKeyPacket) %PGPData.ANY;)
>
<!ELEMENT PGPKeyPacket (#PCDATA)>
<!ELEMENT PGPKeyID (#PCDATA)>
<!-- SPKI Data -->
<!ELEMENT SPKIData (SPKISexp %SPKIData.ANY;) >
<!ELEMENT SPKISexp (#PCDATA)>
<!-- Extensible Content -->
<!ELEMENT Object (#PCDATA|Signature|SignatureProperties|Manifest
%Object.ANY;)* >
<!ATTLIST Object
    Id ID #IMPLIED
    MimeType CDATA #IMPLIED
    Encoding CDATA #IMPLIED
>
<!ELEMENT Manifest (Reference+)>
<!ATTLIST Manifest
    Id ID #IMPLIED
>
<!ELEMENT SignatureProperties (SignatureProperty+)>
<!ATTLIST SignatureProperties
    Id ID #IMPLIED
>
<!ELEMENT SignatureProperty (#PCDATA %SignatureProperty.ANY;)* >
<!ATTLIST SignatureProperty
Target CDATA #REQUIRED
    Id ID #IMPLIED
>
<!-- Algorithm Parameters -->
<!ELEMENT HMACOutputLength (#PCDATA)>
<!ELEMENT DSAKeyValue ((P, Q)?, G?, Y, J?, (Seed, PgenCounter)?)>
<!ELEMENT P (#PCDATA)>
<!ELEMENT Q (#PCDATA)>
<!ELEMENT G (#PCDATA)>
<!ELEMENT Y (#PCDATA)>
<!ELEMENT J (#PCDATA)>
<!ELEMENT Seed (#PCDATA)>
<!ELEMENT PgenCounter (#PCDATA)>
<!ELEMENT RSAKeyValue (Modulus, Exponent)>
<!ELEMENT Modulus (#PCDATA)>
<!ELEMENT Exponent (#PCDATA)>
```

User Registration DTD

```
<?xml version="1.0" encoding="UTF-8"?>

<!ELEMENT Message ((Request, Signature) | Response | (GetResponse, Signature)
| GetProgress | Progress)>
<!ATTLIST Message
    Id CDATA #IMPLIED
>
<!ELEMENT Request (FinalReg | PreReg | CancelReg | UpdateReg)>
<!ATTLIST Request
    IssuerId NMTOKEN #REQUIRED
    Id ID #REQUIRED
>
<!ELEMENT Response (Code, ErrorMessage, ErrorDetail, Warning*)>
<!ELEMENT GetResponse (IssuerId)>
<!ELEMENT GetProgress (IssuerId)>
<!ELEMENT PreReg (DataFormat*, UserReg+)>
<!ELEMENT FinalReg (UserReg+)>
<!ELEMENT UpdateReg (UserUpdate+)>
<!ELEMENT CancelReg (User+)>
<!ELEMENT UserReg (Name?, Password?, Data*, Device*)>
<!ATTLIST UserReg
    Username CDATA #REQUIRED
>
<!ELEMENT UserUpdate (Name?, Username?, Password?)>
<!ELEMENT Device (DeviceType, SerialNo?, Param*)>
<!ATTLIST UserUpdate
    Username CDATA #REQUIRED
>
<!ATTLIST User
    Username CDATA #REQUIRED
>
<!ATTLIST Data
    Name NMTOKEN #REQUIRED
    Value CDATA #REQUIRED
>
<!ATTLIST DataFormat
    Name NMTOKEN #REQUIRED
    Label CDATA #REQUIRED
    Desc CDATA #IMPLIED
    MaxLen NMTOKEN #IMPLIED
    Type (date | string | number | hidden) #IMPLIED
    Format (YYYYMMDD | YYYYMM) #IMPLIED
    Mask (Yes | yes | No | no) #IMPLIED
>
<!ATTLIST Param
    Name NMTOKEN #REQUIRED
>
```

```
<!ELEMENT IssuerId (#PCDATA)>
<!ELEMENT Progress (#PCDATA)>
<!ELEMENT Code (#PCDATA)>
<!ELEMENT ErrorDetail (#PCDATA)>
<!ELEMENT ErrorMessage (#PCDATA)>
<!ELEMENT Warning (#PCDATA)>
<!ELEMENT User EMPTY>
<!ELEMENT Name (#PCDATA)>
<!ELEMENT Username (#PCDATA)>
<!ELEMENT Password (#PCDATA)>
<!ELEMENT Data EMPTY>
<!ELEMENT DataFormat EMPTY>
<!ELEMENT DeviceType (#PCDATA)>
<!ELEMENT SerialNo (#PCDATA)>
<!ELEMENT Param (#PCDATA)>
<!-- DTD for XML Signatures http://www.w3.org/2000/09/xmldsig# -->
<!ENTITY % Object.ANY ''>
<!ENTITY % Method.ANY ''>
<!ENTITY % Transform.ANY ''>
<!ENTITY % SignatureProperty.ANY ''>
<!ENTITY % KeyInfo.ANY ''>
<!ENTITY % KeyValue.ANY ''>
<!ENTITY % PGPData.ANY ''>
<!ENTITY % X509Data.ANY ''>
<!ENTITY % SPKIData.ANY ''>
<!-- Start Core Signature declarations, these should NOT be altered -->
<!ELEMENT Signature (SignedInfo, SignatureValue, KeyInfo?, Object*)>
<!ATTLIST Signature
    xmlns CDATA #FIXED "http://www.w3.org/2000/09/xmldsig#"
    Id ID #IMPLIED
>
<!ELEMENT SignatureValue (#PCDATA)>
<!ATTLIST SignatureValue
    Id ID #IMPLIED
>
<!ELEMENT SignedInfo (CanonicalizationMethod, SignatureMethod, Reference+)>
<!ATTLIST SignedInfo
    Id ID #IMPLIED
>
<!ELEMENT CanonicalizationMethod (#PCDATA %Method.ANY;)* >
<!ATTLIST CanonicalizationMethod
    Algorithm CDATA #REQUIRED
>
<!ELEMENT SignatureMethod (#PCDATA|HMACOutputLength %Method.ANY;)* >
<!ATTLIST SignatureMethod
    Algorithm CDATA #REQUIRED
>
<!ELEMENT Reference (Transforms?, DigestMethod, DigestValue)>
<!ATTLIST Reference
    Id ID #IMPLIED
    URI CDATA #IMPLIED
```

```
    Type CDATA #IMPLIED
>
<!ELEMENT Transforms (Transform+)>
<!ELEMENT Transform (#PCDATA|XPath %Transform.ANY;)* >
<!ATTLIST Transform
    Algorithm CDATA #REQUIRED
>
<!ELEMENT XPath (#PCDATA)>
<!ELEMENT DigestMethod (#PCDATA %Method.ANY;)* >
<!ATTLIST DigestMethod
    Algorithm CDATA #REQUIRED
>
<!ELEMENT DigestValue (#PCDATA)>
<!ELEMENT KeyInfo (#PCDATA|KeyName|KeyValue|RetrievalMethod|
X509Data|PGPData|SPKIData|MgmtData %KeyInfo.ANY;)* >
<!ATTLIST KeyInfo
    Id ID #IMPLIED
>
<!-- Key Information -->
<!ELEMENT KeyName (#PCDATA)>
<!ELEMENT KeyValue (#PCDATA|DSAKeyValue|RSAKeyValue %KeyValue.ANY;)* >
<!ELEMENT MgmtData (#PCDATA)>
<!ELEMENT RetrievalMethod (Transforms?)>
<!ATTLIST RetrievalMethod
    URI CDATA #REQUIRED
    Type CDATA #IMPLIED
>
<!-- X.509 Data -->
<!ELEMENT X509Data ((X509IssuerSerial | X509SKI | X509SubjectName |
X509Certificate | X509CRL )+ %X509Data.ANY;)>
<!ELEMENT X509IssuerSerial (X509IssuerName, X509SerialNumber)>
<!ELEMENT X509IssuerName (#PCDATA)>
<!ELEMENT X509SubjectName (#PCDATA)>
<!ELEMENT X509SerialNumber (#PCDATA)>
<!ELEMENT X509SKI (#PCDATA)>
<!ELEMENT X509Certificate (#PCDATA)>
<!ELEMENT X509CRL (#PCDATA)>
<!-- PGPData -->
<!ELEMENT PGPData ((PGPKeyID, PGPKeyPacket?) | (PGPKeyPacket) %PGPData.ANY;)
>
<!ELEMENT PGPKeyPacket (#PCDATA)>
<!ELEMENT PGPKeyID (#PCDATA)>
<!-- SPKI Data -->
<!ELEMENT SPKIData (SPKISexp %SPKIData.ANY;) >
<!ELEMENT SPKISexp (#PCDATA)>
<!-- Extensible Content -->
<!ELEMENT Object (#PCDATA|Signature|SignatureProperties|Manifest
%Object.ANY;)* >
<!ATTLIST Object
    Id ID #IMPLIED
    MimeType CDATA #IMPLIED
```

```
    Encoding CDATA #IMPLIED
>
<!ELEMENT Manifest (Reference+)>
<!ATTLIST Manifest
    Id ID #IMPLIED
>
<!ELEMENT SignatureProperties (SignatureProperty+)>
<!ATTLIST SignatureProperties
    Id ID #IMPLIED
>
<!ELEMENT SignatureProperty (#PCDATA %SignatureProperty.ANY;)* >
<!ATTLIST SignatureProperty
    Target CDATA #REQUIRED
    Id ID #IMPLIED
>
<!-- Algorithm Parameters -->
<!ELEMENT HMACOutputLength (#PCDATA)>
<!ELEMENT DSAKeyValue ((P, Q)?, G?, Y, J?, (Seed, PgenCounter)?)>
<!ELEMENT P (#PCDATA)>
<!ELEMENT Q (#PCDATA)>
<!ELEMENT G (#PCDATA)>
<!ELEMENT Y (#PCDATA)>
<!ELEMENT J (#PCDATA)>
<!ELEMENT Seed (#PCDATA)>
<!ELEMENT PgenCounter (#PCDATA)>
<!ELEMENT RSAKeyValue (Modulus, Exponent)>
<!ELEMENT Modulus (#PCDATA)>
<!ELEMENT Exponent (#PCD)>
```

Notification DTD

```
<?xml version="1.0" encoding="UTF-8"?>

<!ELEMENT Message ((NotificationRequest, Signature) | (NotificationResponse |
Error))>
<!ATTLIST Message
    Id ID #IMPLIED
>
<!ELEMENT NotificationRequest (StartDate?, EndDate?)>
<!ATTLIST NotificationRequest
    Id ID #REQUIRED
    IssuerId NMTOKEN #IMPLIED
    GroupId NMTOKEN #IMPLIED
    Type (cardreg | cardlock | cardoptout | carddeviceupdate) #REQUIRED
    EncVectorIV CDATA #IMPLIED
>
<!ELEMENT StartDate (#PCDATA)>
<!ELEMENT EndDate (#PCDATA)>
```

```
<!ELEMENT NotificationResponse ((CardLock*) | (CardOptOut*) | (CardReg*)
| (CardDeviceUpdate))>
<!ATTLIST NotificationResponse
    Id ID #REQUIRED
    IssuerId NMTOKEN #IMPLIED
    GroupId NMTOKEN #IMPLIED
    Type (cardreg | cardlock | cardoptout | carddeviceupdate) #REQUIRED
    EncVectorIV CDATA #IMPLIED
>
<!ELEMENT Error (Code, ErrorMessage, ErrorDetail)>
<!ELEMENT Code (#PCDATA)>
<!ELEMENT ErrorMessage (#PCDATA)>
<!ELEMENT ErrorDetail (#PCDATA)>
<!ELEMENT CardReg (Card+)>
<!ELEMENT Card ((Data*, Lock*, Unlock*) | (Data*, Unlock*, Lock*) |
(Data*, Device*, Register?, Confirm?) | (Data*, Device*, Confirm?,
Register?) | (Data*, Optout+) | (Data*, DeviceUpdate+))>
<!ELEMENT CardDeviceUpdate (Card+)>
<!ELEMENT DeviceUpdate EMPTY>
<!ATTLIST DeviceUpdate
    SerialId CDATA #REQUIRED
    DeviceType (VASCO | SMS | EMAIL | OOB | DECOUPLED)
    #REQUIRED
    Action (RegisterNewDevice | RegisterExistingDevice | ActivateByNewDevice |
    ActivateByExistingDevice | UnregisterDevice | RemoveDevice ) #REQUIRED
    Date NMTOKEN #REQUIRED
>
<!ATTLIST Card
    Type (VbV | SPA | JCB | SK | DC) #REQUIRED
    Number NMTOKEN #REQUIRED
    Name CDATA #REQUIRED
>
<!ELEMENT Optout (#PCDATA)>
<!ATTLIST Optout
    Number NMTOKEN #REQUIRED
>
<!ELEMENT Lock (#PCDATA)>
<!ELEMENT Unlock (#PCDATA)>
<!ELEMENT Register (#PCDATA)>
<!ELEMENT Confirm (#PCDATA)>
<!ELEMENT CardLock (Card+)>
<!ELEMENT CardOptOut (Card+)>
<!ELEMENT Data EMPTY>
<!ATTLIST Data
    Name NMTOKEN #REQUIRED
    Value CDATA #REQUIRED
>
<!ELEMENT Device (#PCDATA)>
<!ATTLIST Device
    DeviceType NMTOKEN #REQUIRED
    SerialId CDATA #REQUIRED
```

```
>
<!-- DTD for XML Signatures http://www.w3.org/2000/09/xmldsig# -->
<!ENTITY % Object.ANY ''>
<!ENTITY % Method.ANY ''>
<!ENTITY % Transform.ANY ''>
<!ENTITY % SignatureProperty.ANY ''>
<!ENTITY % KeyInfo.ANY ''>
<!ENTITY % KeyValue.ANY ''>
<!ENTITY % PGPData.ANY ''>
<!ENTITY % X509Data.ANY ''>
<!ENTITY % SPKIData.ANY ''>
<!-- Start Core Signature declarations, these should NOT be altered -->
<!ELEMENT Signature (SignedInfo, SignatureValue, KeyInfo?, Object*)>
<!ATTLIST Signature
    xmlns CDATA #FIXED "http://www.w3.org/2000/09/xmldsig#"
    Id ID #IMPLIED
>
<!ELEMENT SignatureValue (#PCDATA)>
<!ATTLIST SignatureValue
    Id ID #IMPLIED
>
<!ELEMENT SignedInfo (CanonicalizationMethod, SignatureMethod, Reference+)>
<!ATTLIST SignedInfo
    Id ID #IMPLIED
>
<!ELEMENT CanonicalizationMethod (#PCDATA %Method.ANY;)* >
<!ATTLIST CanonicalizationMethod
    Algorithm CDATA #REQUIRED
>
<!ELEMENT SignatureMethod (#PCDATA|HMACOutputLength %Method.ANY;)* >
<!ATTLIST SignatureMethod
    Algorithm CDATA #REQUIRED
>
<!ELEMENT Reference (Transforms?, DigestMethod, DigestValue)>
<!ATTLIST Reference
    Id ID #IMPLIED
    URI CDATA #IMPLIED
    Type CDATA #IMPLIED
>
<!ELEMENT Transforms (Transform+)>
<!ELEMENT Transform (#PCDATA|XPath %Transform.ANY;)* >
<!ATTLIST Transform
    Algorithm CDATA #REQUIRED
>
<!ELEMENT XPath (#PCDATA)>
<!ELEMENT DigestMethod (#PCDATA %Method.ANY;)* >
<!ATTLIST DigestMethod
    Algorithm CDATA #REQUIRED
>
<!ELEMENT DigestValue (#PCDATA)>
<!ELEMENT KeyInfo (#PCDATA|KeyName|KeyValue|RetrievalMethod|
```

```
X509Data|PGPData|SPKIData|MgmtData %KeyInfo.ANY;)* >
<!ATTLIST KeyInfo
    Id ID #IMPLIED
>
<!-- Key Information -->
<!ELEMENT KeyName (#PCDATA)>
<!ELEMENT KeyValue (#PCDATA|DSAKeyValue|RSAKeyValue %KeyValue.ANY;)* >
<!ELEMENT MgmtData (#PCDATA)>
<!ELEMENT RetrievalMethod (Transforms?)>
<!ATTLIST RetrievalMethod
    URI CDATA #REQUIRED
    Type CDATA #IMPLIED
>
<!-- X.509 Data -->
<!ELEMENT X509Data ((X509IssuerSerial | X509SKI | X509SubjectName |
X509Certificate | X509CRL )+ %X509Data.ANY;)>
<!ELEMENT X509IssuerSerial (X509IssuerName, X509SerialNumber)>
<!ELEMENT X509IssuerName (#PCDATA)>
<!ELEMENT X509SubjectName (#PCDATA)>
<!ELEMENT X509SerialNumber (#PCDATA)>
<!ELEMENT X509SKI (#PCDATA)>
<!ELEMENT X509Certificate (#PCDATA)>
<!ELEMENT X509CRL (#PCDATA)>
<!-- PGPData -->
<!ELEMENT PGPData ((PGPKeyID, PGPKeyPacket?) | (PGPKeyPacket) %PGPData.ANY;)
>
<!ELEMENT PGPKeyPacket (#PCDATA)>
<!ELEMENT PGPKeyID (#PCDATA)>
<!-- SPKI Data -->
<!ELEMENT SPKIData (SPKISexp %SPKIData.ANY;) >
<!ELEMENT SPKISexp (#PCDATA)>
<!-- Extensible Content -->
<!ELEMENT Object (#PCDATA|Signature|SignatureProperties|Manifest
%Object.ANY;)* >
<!ATTLIST Object
    Id ID #IMPLIED
    MimeType CDATA #IMPLIED
    Encoding CDATA #IMPLIED
>
<!ELEMENT Manifest (Reference+)>
<!ATTLIST Manifest
    Id ID #IMPLIED
>
<!ELEMENT SignatureProperties (SignatureProperty+)>
<!ATTLIST SignatureProperties
    Id ID #IMPLIED
>
<!ELEMENT SignatureProperty (#PCDATA %SignatureProperty.ANY;)* >
<!ATTLIST SignatureProperty
    Target CDATA #REQUIRED
    Id ID #IMPLIED
```

```
>
<!-- Algorithm Parameters -->
<!ELEMENT HMACOutputLength (#PCDATA)>
<!ELEMENT DSAKeyValue ((P, Q)?, G?, Y, J?, (Seed, PgenCounter)?)>
<!ELEMENT P (#PCDATA)>
<!ELEMENT Q (#PCDATA)>
<!ELEMENT G (#PCDATA)>
<!ELEMENT Y (#PCDATA)>
<!ELEMENT J (#PCDATA)>
<!ELEMENT Seed (#PCDATA)>
<!ELEMENT PgenCounter (#PCDATA)>
<!ELEMENT RSAKeyValue (Modulus, Exponent)>
<!ELEMENT Modulus (#PCDATA)>
<!ELEMENT Exponent (#PCDATA)>
```

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 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[^2].

  [^2]: 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.

Remote Messaging

This section details the messaging requirements for connecting between ActiveAccess and an issuer's remote systems.

The ActiveAccess authentication system is responsible for managing the authentication of cardholders during American Express SafeKey, Diners Club International ProtectBuy, JCB J/Secure, Mastercard SecureCode / Identity Check and Verified by Visa / Visa Secure transactions. In order to support this requirement, the system must have access to appropriate information in order to uniquely determine the identity of the cardholder, whether a cardholder transaction requires authentication and what type of authentication is required. The determination of whether a cardholder is registered, whether a transaction requires authentication and what type of authentication is required for any particular transaction is currently performed by the ActiveAccess system. However, in order to delegate any of these duties to external issuer systems, some level of integration will be required.

In order to determine the correct 3-D Secure registration status, an issuer may be required to maintain the status of each of its cardholder's within the ActiveAccess system. Where this requires a significant investment in the development of maintenance procedures, an alternative may be to connect to determine the registration status of a cardholder by connecting with an issuer's systems. This procedure will remove the need to synchronise systems and ensure the maintenance of only one source of truth.

In a similar way, where an issuer is currently providing authentication services for its cardholders and wishes to re-use some of these services, it is possible to connect to the issuer's existing authentication system and leverage off their existing process for cardholder authentication. While the capability to perform second factor authentication is provided by ActiveAccess, this integration may be the issuer's preferred implementation model. This approach ensures a seamless user experience for the bank's customers across its banking channels.

The following sections explain the messaging requirements for connecting between ActiveAccess and an issuer's remote systems.

Message Format

SOAP, originally defined as Simple Object Access Protocol, is a protocol specification for exchanging structured information in the implementation of Web Services for messaging between ActiveAccess and an external system. It relies on Extensible Markup Language (XML) for its message format, and usually relies on other Application Layer protocols, most notably Hypertext Transfer Protocol (HTTP) and Simple Mail Transfer Protocol (SMTP), for message negotiation and transmission.

The Web Services Description Language is an XML-based language that is used for describing the functionality offered by a Web service. A WSDL description of a web service (also referred to as a WSDL file) provides a machine-readable description of how the service can be called, what parameters it expects, and what data structures it returns. It thus serves a roughly similar purpose as a method signature in a programming language.

Support for generating client-side and server-side API code based on WSDL is provided in most languages.

For WSDL of the services discussed in this document, refer to **Remote System Integration WSDL**.

Request

During a 3-D Secure transaction, the first stage of the messaging is to determine the registration status of a cardholder. By integrating with the issuer's system, ActiveAccess can determine the cardholder's registration status and therefore determine whether a transaction requires authentication.

Having determined a transaction requires authentication, the second stage in the process is to perform an authentication and by integrating with the issuer's system, ActiveAccess is capable of reusing the issuer's infrastructure to determine the authentication result. At the end of the process, the ActiveAccess system responds to the MPI with the authentication result in accordance with the 3-D Secure protocol.

The purpose of remote system integration is to:

- Determine the registration status of a cardholder

  and/or

- Initiate and verify the cardholder authentication.

The types of messages sent by ActiveAccess are:

- **VerifyRegistration**: determine the registration status of a cardholder
- **InitAuthentication**: initiate the cardholder authentication process
- **VerifyAuthentication**: verify the authentication result
- **PreAuthentication**: determine the action for exemption
- **VerifyIdentity**: verify the identification results
- **Register**: register the card
- **ResetPassword**: initiate the reset password process
- **Ping**: determine the status of the service.

> **Note**
>
> Messages sent between ActiveAccess and the remote system for this purpose do not carry any session information and therefore are considered to be stateless

CAAS Services

Table 1 - CAAS Services

CAAS Service	Table 1	
Operation	**Description**	**Usage**
VerifyRegistration	Used to verify the registration status of a cardholder.	Required for a verify registration request
InitAuthentication	Used to initiate the authentication process for out-of-band authentication.	Required for an initiate authentication request
VerifyAuthentication	Used to determine the authentication result.	Required for a verify authentication request
PreAuthentication	Used to determine the action for exemption	Optional

CAAS Service	Table 1	
VerifyIdentity	Used to verify the identification results	Required for a reset password request and register request
Register	Used to register the card	Required for a register request
ResetPassword	Used to initiate the reset password process	Required for a reset password request
Ping	Used to determine if service is up and running	Optional

Verify Registration

The Verify Registration request is used to determine the registration status of a cardholder, within the remote system. Where a cardholder cannot be uniquely identified, such as the case where primary and secondary exist, it may be necessary for the remote system to provide the registration status of all related cardholders. ActiveAccess will then determine the appropriate course of action based on the response and in line with the issuer's business requirements.

Once the cardholder has been uniquely identified and where authentication is required, ActiveAccess should commence the appropriate authentication process.

Verify Registration Request

Table 2 - VerifyRegReq

VerifyRegReq	Table 2		
Attribute	**Description**	**Usage**	Sample Value
Card	Refer to *Table 3 - Card*	Required	
Transaction	Additional transaction information may include transaction; cardholder and merchant information such as **MerchantID** and **AcqBIN**. Refer to *Table 4 - Transaction*.	Optional	

VerifyRegReq	Table 2		
IV	If an encryption KeyStore has been defined for the issuer or group of issuers, critical card data must be encrypted by ActiveAccess using DESede/CBC/PKCS5Padding mode. The CBC encryption mode requires an Initialisation Vector (IV), which includes 8 random bytes, as an input parameter for encryption and decryption. The IV should be sent to the CAAS server to be used at decryption time. To do this, the IV which was used for encryption, must be encrypted in DESede/ECB/ PKCS5Padding mode, using the same key, then HEX encoded and set as IV in the request.	Optional- If present, it means that ActiveAccess has generated an IV parameter and critical card information has been encrypted using the CBC mode and the generated IV, otherwise ECB or plain mode has been used instead. 16 bytes when AES, 8 bytes when DESede.	8F51F71064DB2B65

Table 3 - Card

Card	Table 3		
Attribute	**Description**	**Usage**	**Sample Value**
ID	A unique cardholder identifier	Optional. Up to 2000 characters.	2345678901
Number	Card number (If an encryption KeyStore has been defined for the issuer or group of issuers, card number will be encrypted by ActiveAccess using DESede/CBC/ PKCS5Padding mode and IV, then HEX encoded and included in the message. Therefore, the CAAS server will need to decrypt this field using DESede/CBC/ PKCS5Padding mode and the request's IV before using it in the process)	Optional. Up to 64 characters.	5012345678901234

Card	Table 3		
CardName	Name on card (If an encryption KeyStore has been defined for the issuer or group of issuers, name on card will be encrypted by ActiveAccess using DESede/CBC/ PKCS5Padding mode and IV, then HEX encoded and included in the message. Therefore, the CAAS server will need to decrypt this field using DESede/CBC/ PKCS5Padding mode and the request's IV before using it in the process)	Optional. Up to 512 characters.	JOE CITIZEN
Type	Card type	Optional. Up to 3 characters. Valid types: VbV – Visa, SPA – Mastercard, JCB – JCB, SK – American Express, DC - Diners Club International.	SPA
Context_Blob	A context detail that may be used in subsequent calls	Optional. This field can be ignored by CAAS in VerifyRegReq as ActiveAccess does not use it and only echoes it in InitAuthReq and VerifyAuthReq if it has been set in VerifyRegResp.CardInfo. Context_Blob by CAAS. Length not defined.	12345678901235467890
LanCode	A code between 0 to 4 that presents the cardholder's preferred language	Optional. 1 character in length.	0

Table 4 - Transaction

Transaction	Table 4		
Attribute	**Description**	**Usage**	**Sample Value**
XID	The transaction ID as defined in the PAReq message	Optional. Up to 28 characters.	MDAwMDAwMDAwMDAwMDAwMDAxMDA=
PurchaseDate	The transaction purchase date and time as defined in the PAReq message	Optional. Up to 17 characters in XMLGregorianCalendar format.	20091023 06:11:00
PurchaseAmount	The transaction purchase amount as defined in the PAReq message	Optional. Up to 12 characters in decimal format.	12345
PurchaseCurrency	The 3-digit transaction currency value as defined in the PAReq message. Refer to **Country and Currency Codes**	Optional. Up to 3 digits.	840
PurchaseExponent	The minor units of currency specified in ISO 4217	Optional. 1 character in length.	2
PurchaseDesc	A description of the purchase as defined in the PAReq message	Optional. Up to 125 characters.	Blue Shirt
MerchantID	The merchant ID as defined in the PAReq message	Optional. Up to 24 characters.	123456789012345

Transaction	Table 4		
AcqBIN	The acquirer BIN as defined in the PAReq message	Optional. Up to 11 characters.	412345
MerchantName	The merchant name as defined in the PAReq message	Optional. Up to 25 characters.	Test Merchant
MerchantURL	The fully qualified merchant URL as defined in the PAReq message	Optional. Up to 2048 characters.	http://www.testmerchant.com.au/
MerchantCountry	The 3-digit merchant country code as defined in the PAReq message. Refer to **Country and Currency Codes**	Optional. 3 digits in length.	036
CardExpiry	The 4-digit expiry date of the card as defined in the PAReq message, e.g. YYMM	Optional. 4 or 6 digits in length.	1012
CardholderIP	The IP address of the cardholder browser where available	Optional. 15 or 45 characters in IPv4 or IPv6 format.	192.168.0.157
CVD	Card Verification Data code is the 3 or 4-digit code found on the back of a payment card	Optional. 3 or 4 digits in length.	0320

Transaction	Table 4		
issuerName	Name of Issuer/Bank to be displayed on OOB page	Optional. Up to 64 characters.	Any Bank
theeDSProtocolVersion	Version of 3DS protocol in x.x.x format	Optional. 5 characters in length.	2.1.0
acsTransId	Universally Unique transaction identifier assigned by the ACS to identify a single transaction.	Optional. 36 alphanumeric characters in length.	ee5de3bc-a1a3-4648-9c5f-350422146fe1
threeDSTransId	Universally Unique transaction identifier assigned by the 3DS Server to identify a single transaction.	Optional. 36 alphanumeric characters in length.	we5de3bc-a213-46lk-9cas-35456ed46fe1
dsTransId	Universally Unique transaction identifier assigned by the Directory Server to identify a single transaction.	Optional. 36 alphanumeric characters in length.	tg6de3bc-a213-4r3k-9c12-35456ed4edr43
threeDSRequestorID	DS assigned 3DS Requestor identifier	Optional. Variable length with maximum 35 characters in length.	tg6de3bc-a213-4r3k-9c12-35456ed4edr43
threeDSRequestorName	DS assigned 3DS Requestor name	Optional. Variable length with maximum 40 characters in length.	Test Requestor
threeDSRequestorURL	URL of 3DS Requestor website or customer care site	Optional. Variable length with maximum 2048 characters in length.	http://server.domainname.com

Transaction	Table 4		
threeDSServerRefNumber	Unique identifier assigned by the EMVCo	Optional. Variable length with maximum 32 characters in length.	TestTDSRef123
threeDSServerOperatorID	DS assigned 3DS Server identifier	Optional. Variable length with maximum 32 characters in length.	TestDsOperatorId12
threeDSServerURL	URL of the 3DS Server to which the DS will send the RReq	Optional. Variable length with maximum 2048 characters in length.	https://server.adomainname.net
deviceChannel	Indicates the type of channel interface being used to initiate the transaction	Optional. 2 characters in length. Acceptable values are 01,02,03	01
dsReferenceNumber	Unique identifier assigned by the EMVCo	Optional. Variable length with maximum 32 characters in length.	TestDsRef123
payTokenInd	A value of True indicates that the transaction was de-tokenised prior to being received by the ACS.	Optional. 4 characters in length. Acceptable value: true	true
purchaseInstalData	Indicates the maximum number of authorisations permitted for instalment payments	Optional. Variable length with maximum 3 characters in length. Acceptable values: number greater than 1	3
mcc	DS-specific code describing the Merchant's type of business, product or service	Optional. 4 characters in length.	3210

Transaction	Table 4		
messageCategory	Identifies the category of the message for a specific use case.	Optional. 2 characters in length. Acceptable values: 01, 02	01
recurringExpiry	Date after which no further authorisations shall be performed.	Optional. 8 characters in length. Format YYYYMMDD	20201010
recurringFrequency	Indicates the minimum number of days between authorisations	Optional. Variable length with maximum 4 characters in length.	3
sdkReferenceNumber	Unique identifier assigned by the EMVCo	Optional. Variable length with maximum 32 characters in length.	TestSDKRef123
transType	Identifies the type of transaction being authenticated.	Optional. 2 characters in length. Acceptable values: 01, 03, 10, 11, 28	01
acctType	Indicates the type of account. For example, for a multi-account card product.	Optional. 2 characters in length. Acceptable values: 01, 02, 03	01

Verify Registration Response

A response message should be sent back for each request. The response message should provide the result of the request message with details of appropriate response information or errors as appropriate. Where one card is found in the remote system, registration details for that card should be included in the response. Where multiple cards are found, the registration details for each of the cards should be included in the response.

Table 5 - VerifyRegResp

VerifyRegResp	Table 5		
Attribute	**Description**	**Usage**	**Sample Value**
CardInfo	If the request was successful and at least one card record was found, card related data may include primary/secondary cardholder indicator, registration status, authentication required indicator, authentication type, a card identifier and a SIS data. Refer to *Table 6 - CardInfo*.	Conditional. If response code is not presented, at least one CardInfo should exist.	
Code	Response code: 0 - request was successful but no card records were found 1 - request has been successfully processed but there are warnings (NOTE- Please see below) 2 - error in processing the request.	Required. Included where no card records are found or an error occurred.	0
ErrorMessage	A descriptive message that identifies the category of the error	Conditional. Included where a Code is returned in the response.	No card(s) found
ErrorDetail	A more detailed description of the error	Conditional. Included where a Code is returned in the response.	No card(s) matching the request were found

Note

ActiveAccess treats warnings (code=1) as errors unless the exact **ErrorMessage** is introduced in **AA_HOME/caaswarning.properties** with a code less than 2000. Changing this file requires a restart to take effect.

Table 6 - CardInfo

CardInfo	Table 6		
Attribute	**Description**	**Usage**	**Sample Value**
CardID	A unique cardholder identifier to be used as the value of the Card.ID attribute in subsequent request messages	Conditional. At least one of the Context_Blob or CardID is required. ActiveAccess echoes the Context_Blob into both Card.ID and Card. Context_Blob of the subsequent InitAuthReq and VerifyAuthReq if no CardID is returned by CAAS	2345678901

CardInfo	Table 6		
Card Name	Cardholder name to be used for specifying the exact cardholder when there are multiple cardholders for an identical card number (If an encryption KeyStore has been defined for the issuer or group of issuers, cardholder name must be encrypted using DESede/CBC/ PKCS5Padding mode and message request IV by CAAS server, then HEX encoded and included in the message. ActiveAccess will decrypt this field using DESede/CBC/ PKCS5Padding mode and the message request IV before using it in the process.)	Optional	John Smith

CardInfo	Table 6		
PAM	Personal Assurance Message (If an encryption KeyStore has been defined for the issuer or group of issuers, PAM must be encrypted using DESede/CBC/ PKCS5Padding mode and the message request IV by the CAAS server, then HEX encoded and included in the message. ActiveAccess will decrypt this field using DESede/ CBC/PKCS5Padding mode and the message request IV before using it in the process.)	Optional	This is my Bank
Context_Blob	A context detail that may be used in subsequent calls	Conditional. At least one of the Context_Blob or CardID is required. ActiveAccess echoes the Context_Blob into both Card.ID and Card. Context_Blob of the subsequent InitAuthReq and VerifyAuthReq if no CardID is returned by CAAS	12345678901234567890

CardInfo	Table 6		
Prisec	Primary or Secondary Cardholder 1 - Primary 2 - Secondary	Conditional	1

CardInfo	Table 6		
RegStatus	Registration Status: 1 - Enrolled (ActiveAccess enrolment status of pre-registered) 2 - Registered (ActiveAccess enrolment status of registered) 3 - Locked 4 - Unknown 5 - Error 6 - Temporarily Exempt 7 - Permanently Exempt 8 - Lost 9 - Stolen 10 -Restricted 11 - Card Number Error 12 - No Account 13 - Fraud 14 - Expired	Conditional. If SIS has data, RegStatus will not be considered.	2

CardInfo	Table 6		
AuthRequired	Authentication Required: 1 - Yes 2 - No	Conditional. If SIS has data, AuthRequired will not be considered.	1
AuthType	Authentication Type: 1 - Password 2 - SMS 3 - OTP device 4 - Virtual OTP device 5 - CAP/DPA 6 - Verify by Voice 7 - USS 8 - Q&A 9 - OLB 10 - CR 11 - BIO 12 - PKI 13 - TTP 14 - Email 15 - OOB	Conditional. If SIS has data, AuthType will not be considered. Note that this field **cannot** be used for two-factor authentication or being selected by user/cardholder during the authentication, as it only allows one supplementary authentication type to be set for the authentication page.	1

CardInfo	Table 6		
RegToken	The variable part of a message to be displayed to user/cardholder in the registration page. It reflects the number of times the cardholder opts-out during the registration process.	Optional. e.g. CAAS server wants to limit the number of times that a user/cardholder can opt-out from the registration process.	3 (e.g. of the message in the registration page: You have opted-out of the registration process 3 times)

CardInfo	Table 6		
AuthTypeSup	Supplementary authentication types that user/cardholder's account supports: 1 - Password 2 - SMS 3 - OTP device 4 - Virtual OTP device 5 - CAP/DPA 6 - Verify by Voice (OOB Biometrics) 7 - USS 8 - Q&A 9 - OLB (OOB Login) 10 - CR 11 - BIO (OOB Biometrics) 12 - PKI 13 - TTP (OOB Other) 14 - Email 15 - OOB (OOB Other)	Optional. Note that only this field can be used for two-factor authentication or being selected by user/cardholder during the authentication, as it allows more than one supplementary authentication type to be set for the authentication page.	2, 3
SIS	Refer to *Table 7 - SIS*	Conditional. If exists, it takes precedence over RegStatus, AuthRequired and AuthType	

CardInfo	Table 6		
ProofAttempt	The availability of the Opt-Out option, as opposed to Cancel, for the cardholder. True - request identification parameters False - Proof of Attempt disabled,Opt - Out option not available **Note** - it is recommended to set this through ACS via **MIA > Issuers > Settings** instead of this parameter	Optional	false

CardInfo	Table 6		
ActivationDuringShopping	The ability to authenticate an enrolled cardholder by ID details for verification. True - request identification parameters False - registration pages are processed as without activation **Note** - it is recommended to set this through ACS via **MIA > Issuers > Settings** instead of this parameter	Optional	true
LanCode	The code of the preferred language saved for the cardholder. The value can be a digit between 0 to 4.	Optional	0 - default language 1 - 2nd language 2 - 3rd language 3 - 4th language 4 - 5th language

IdentityData	Attributes of Data: **Name** (required) - the name of the AuthData parameter to be used for data collection on the page **AuthType** (conditional) - empty value Format (optional) - the regular expression for verifying the value collected from the page **Mask** (optional) - True - input on the page will be masked False - input on the page will be in plaintext **Confirm** (optional) - True - an additional input field will be added to the page for confirmation False - no confirmation input field will be displayed on the page Refer to Table 8 - Data	Conditional. Required if ActivationDuringShopping is TRUE and RegStatus is 1. If RegStatus is not 1 and IdentityData has been returned, it will be used in the ResetPassword process.	identityData=[data={[value=<(null)>, error=<(null)>, name=pin, authType=<(null)>, format=\w+, mask=true, confirm=<(null)>]

CardInfo	Table 6		
twoFA	The availability of 2FA authentication option. 2FA authentication is a combination of: **Knowledge**: something only the user knows (e.g. password, pin, ID number) + **Ownership**: something only the user possesses (e.g. mobile device, token, smart card) or **Inherence**: something only the user is (e.g. fingerprint, face or voice recognition) The first factor must be knowledge; the second factor can be ownership or inherence.	Optional	true - enable two-factor authentication false - disable two factor authentication

Table 7 - SIS

SIS	Table 7		
Attribute	**Description**	**Usage**	**Sample Value**
AccountState	Account State: 1 - Operational 2 - Unknown	Required	1
OperationalState	Operational State: 1 - Operational 2 - Locked Blank -Not Specified	Required	1
SecurityDeviceType	Security Device Type: 1 - Hard Token 2 - Soft Token 3 - SMS 4 - PIQ 5 - Email Blank - Not specified	Required	3
IsExempt	Authentication Exemption: True False Blank - Not specified	Required	2
IsPermanent	Permanent Authentication Exemption: True False Blank - Not specified	Required	2

Table 8 - Data

Data	Table 8		
Attribute	**Description**	**Usage**	**Sample Value**
Value	The value of Data	Optional	123456
Error	Refer to *Table 9* - Error	Optional	

Table 9 - Error

Error	Table 9		
Attribute	**Description**	**Usage**	**Sample Value**
Code	Response code: 0 - the request was successful 1 - there was an error and ActiveAccess should send the request again 2 - there was an error and ActiveAccess should cancel the authentication	Required	2
Message	A descriptive message that identifies the category of the error	Optional	No card(s) found
Detail	A more detailed description of the error	Optional	No card(s) matching the request were found

Pre Authentication

The ability to integrate ActiveAccess with an external risk engine has been established in the Pre Authentication process in which header data including cookie and HTTP header data in addition to potential extension information will be sent to CAAS for it to determine if authentication is required or exempt.

Pre Authentication

Table 10 - PreAuthReq

PreAuthReq	Table 10		
Attribute	**Description**	**Usage**	**Sample Value**
Card	Where a value for Card.ID or Context_Blob was returned in the VerifyReg response, this value should be assigned to the Card.ID attribute. Otherwise, attributes of the Card may include **Number, Name** and **Type** as described in the VerifyReg request Refer to *Table 3 - Card*.	Required Either ID or Number and Type should be presented	card=[id=4564260131003313, number=4564-26XX-XXXX-3313, type=VbV, cardName=<(null)>, Context_Blob=595],
Transaction	Where messaging commences after the ActiveAccess system receives the PAReq, additional transaction, cardholder and merchant information is available. This information may be additionally sent to the issuer system for analysis and fraud detection purposes. Where required, the following data fields may be sent to the issuer's system in any of the request messages. Refer to *Table 4 - Transaction*.	Optional	transaction=[xid=MDAwMDAwMDAwMDAwMDAwMDAxMDA=, purchaseAmount=12365, purchaseCurrency=840, purchaseDate=[eon=<(null)>, year=2016, month=11, day=3, timezone=210, hour=10, minute=16, second=46, fractionalSecond=0.000],

PreAuthReq	Table 10		
IV	If an encryption KeyStore has been defined for the issuer or group of issuers, critical card data must be encrypted by ActiveAccess using DESede/CBC/PKCS5Padding mode. The CBC encryption mode requires an Initialisation Vector (IV), which includes 8 random bytes, as an input parameter for encryption and decryption. The IV should be sent to the CAAS server to be used at decryption time. To do this, the IV which was used for encryption, must be encrypted in DESede/ECB/PKCS5Padding mode, using the same key, then HEX encoded and set as IV in the request.	Optional. If present, it means that ActiveAccess has generated an IV parameter and critical card information has been encrypted using the CBC mode and the generated IV, otherwise ECB or plain mode has been used instead.	8F51F71064DB2B65

PreAuthReq	Table 10		
HeaderParams	Attributes of Param: **Value** (required) **Key** (required) **Cookie** (optional)	Optional	headerParams=[param={[value=value1, key=key1, cookie=true],...}],
ExtensionParams	Attributes of Param: **Value** (required) **Key** (required)	Optional	extensionParams =[param={[value=value1, key=key1],...}],
AdditionalParams	Attributes of Param: **Value** (required) **Key** (required)	Optional	additionalParams =[param={[value=50, key=giftCardAmount],...}],

Table 11 - HeaderParams

HeaderParams	Table 11		
Attribute	**Description**	**Sample Value**	
User-Agent	Either value of HTTP request header parameter or browserUserAgent element of AReq; Exact content of the HTTP user-agent header	Mozilla/5.0 (X11; Linux x86_64; rv:12.0) Gecko/ 20100101 Firefox/12.0	
Accept	Either value of HTTP request header parameter or browserAcceptHeader element of AReq; Exact content of the HTTP accept headers	text/html	
Accept-Language	Either the value of HTTP request header parameter or browserLanguage element of AReq	en-US	
proxy-ip	Either the value of HTTP request header parameter or browserIp element of AReq; IP address of the browser as returned by the HTTP headers	192.168.1.138	
browserJavaEnabled	browserJavaEnabled element of AReq; Boolean that represents the ability of the cardholder browser to execute Java. Acceptable values: true, false	true	
browserTZ	browserTZ element of AReq; Time difference between UTC time and the Cardholder browser local time, in minutes	--	
browserLanguage	browserLanguage element of AReq; Value representing the browser language	en-US	
deviceInfo	deviceInfo element of AReq; Device information gathered by the 3DS SDK from a Consumer Device	--	
sdkAppID	sdkAppID element of AReq; Universally unique ID created upon all installations and updates of the 3DS Requestor App on a Consumer Device.	--	
browserColorDepth	browserColorDepthelement of AReq; Value representing the bit depth of the colour palette for displaying images, in bits per pixel. Acceptable values: 1, 4, 8, 15, 16, 24, 32, 48	15	

HeaderParams	Table 11		
browserScreenHeight	browserScreenHeight of AReq; Total height of the Cardholder's screen in pixels, 1 - 6 characters in length	390	
browserScreenWidth	browserScreenWidth of AReq; Total width of the cardholder's screen in pixels, 1 - 6 characters in length	400	

ExtensionParams

The elements differ by different message extensions which are defined in messages. For instance, American Express extension params differ from Mastercard extension params.

Table 12 - AdditionalParams

AdditionalParams	Table 12		
Attribute	**Description**	**Sample Value**	
shipAddrState	shipAddrState element of AReq; The state or province of the shipping address associated with the card being used for this purchase, maximum 3 characters in length. Value accepted: Should be the country subdivision code defined in ISO 3166-2	--	
shipAddrCity	shipAddrCity element of AReq; City portion of the shipping address requested by the Cardholder. Maximum 50 characters in length	--	
shipAddrCountry	shipAddrCountry element of AReq; Country of the shipping address requested by the Cardholder. 3 characters on length. Value accepted: ISO 3166-1 three-digit country code	--	
shipAddrLine1	shipAddrLine1 element of AReq; First line of the street address or equivalent local portion of the shipping address requested by the Cardholder. Maximum 50 characters in length	--	

AdditionalParams	Table 12		
shipAddrLine2	shipAddrLine2 element of AReq; The second line of the street address or equivalent local portion of the shipping address requested by the Cardholder. Maximum 50 characters in length	--	
shipAddrLine3	shipAddrLine3 element of AReq; The third line of the street address or equivalent local portion of the shipping address requested by the Cardholder. Maximum 50 characters on length	--	
shipAddrPostCode	shipAddrPostCode element of AReq; The ZIP or other postal code of the shipping address requested by the Cardholder. Maximum 16 characters in length	--	
billAddrState	billAddrState element of AReq; The state or province of the Cardholder billing address associated with the card used for this purchase. Maximum 3 characters in length. Value accepted: Should be the country subdivision code defined in ISO 3166-2	--	
billAddrCity	billAddrCity element of AReq; The city of the Cardholder billing address associated with the card used for this purchase. Maximum 50 characters.	--	
billAddrCountry	billAddrCountry element of AReq; The country of the Cardholder billing address associated with the card used for this purchase. 3 characters in length. Value accepted: Shall be the ISO 3166-1 numeric three-digit country code	--	
billAddrLine1	billAddrLine1 element of AReq; First line of the street address or equivalent local portion of the Cardholder billing address associated with the card used for this purchase. Maximum 50 characters in length	--	
billAddrLine2	billAddrLine2 element of AReq; Second line of the street address or equivalent local portion of the Cardholder billing address associated with the card used for this purchase. Maximum 50 characters	--	

AdditionalParams	Table 12		
billAddrLine3	billAddrLine3 element of AReq; Third line of the street address or equivalent local portion of the Cardholder billing address associated with the card used for this purchase. Maximum 50 characters	--	
billAddrPostCode	billAddrPostCode element of AReq; ZIP or other postal code of the Cardholder billing address associated with the card used for this purchase. Maximum 16 characters	--	
deliveryEmailAddress	deliveryEmailAddress element of AReq; For Electronic delivery, the email address to which the merchandise was delivered. Maximum 254 characters	--	
deliveryTimeframe	deliveryTimeframe element of AReq; Indicates the merchandise delivery timeframe. 2 characters in length. Values accepted: 01, 02, 03, 04	--	
giftCardAmount	giftCardAmount element of AReq; For prepaid or gift card purchase, the purchase amount total of prepaid or gift card(s) in major units (for example, USD 123.45 is 123) Maximum 15 characters	123	
giftCardCount	giftCardCount element of AReq; For prepaid or gift card purchase, total count of individual prepaid or gift cards/codes purchased. 2 charatsers	--	
giftCardCurr	giftCardCurr element of AReq; For prepaid or gift card purchase, the currency code of the card as defined in ISO 4217. 3 characters	--	
preOrderDate	preOrderDate element of AReq; For a pre-ordered purchase, the expected date that the merchandise will be available. Accepted format: YYYYMMDD	--	
preOrderPurchaseInd	preOrderPurchaseInd element of AReq; Indicates whether Cardholder is placing an order for merchandise with a future availability or release date. 2 characters. Value accepted: 01, 02	--	

AdditionalParams	Table 12		
reorderItemsInd	reorderItemsInd element of AReq; Indicates whether the cardholder is reordering previously purchased merchandise. 2 characters. Value Accepted: 01, 02	--	
shipIndicator	shipIndicator element of AReq; Indicates shipping method chosen for the transaction. 2 characters. Value accepted: 01, 02, 03, 04, 05, 06, 07	--	
threeDSReqAuthData	threeDSReqAuthData element of AReq; Data that documents and supports a specific authentication process. Maximum 2048 characters	--	
threeDSReqAuthMethod	threeDSReqAuthMethod element of AReq; Mechanism used by the Cardholder to authenticate to the 3DS Requestor. 2 characters. Value accepted: 01, 02, 03, 04, 05, 06	--	
threeDSReqAuthTimestamp	threeDSReqAuthTimestamp element of AReq; Date and time in UTC of the cardholder authentication. 12 characters. Format accepted: YYYYMMDDHHMM	--	
threeDSReqPriorAuthData	threeDSReqPriorAuthData element of AReq; Data that documents and supports a specific authentication process. Maximum 2048 characters	--	
threeDSReqPriorAuthMethod	threeDSReqPriorAuthMethod element of AReq; Mechanism used by the Cardholder to previously authenticate to the 3DS Requestor. 2 characters. Value accepted: 01, 02, 03, 04	--	
threeDSReqPriorAuthTimestamp	threeDSReqPriorAuthTimestamp element of AReq; Date and time in UTC of the prior cardholder authentication. 12 characters. Format accepted: YYYYMMDDHHMM	--	
threeDSReqPriorRef	threeDSReqPriorRef element of AReq; This data element provides additional information to the ACS to determine the best approach for handling a request. 36 characters	--	

AdditionalParams	Table 12		
chAccAgeInd	chAccAgeInd element of AReq; Length of time that the cardholder has had the account with the 3DS Requestor. 2 characters. Value accepted: 01, 02, 03, 04, 05	--	
chAccChange	chAccChange element of AReq; Date that the cardholder's account with the 3DS Requestor was last changed, including Billing or Shipping address, new payment account, or new user(s) added. 8 characters. Format accepted: YYYYMMDD	--	
chAccChangeInd	chAccChangeInd element of AReq; Length of time since the cardholder's account information with the 3DS Requestor was last changed, including Billing or Shipping address, new payment account, or new user(s) added. 2 characters. Value accepted: 01, 02, 03, 04	--	
chAccDate	chAccDate element of AReq; Date that the cardholder opened the account with the 3DS Requestor. 8 characters. Format accepted: YYYYMMDD	--	
chAccPwChange	chAccPwChange element of AReq; Date that cardholder's account with the 3DS Requestor had a password change or account reset. 8 characters. Format accepted: YYYYMMDD	--	
chAccPwChangeInd	chAccPwChangeInd element of AReq; Indicates the length of time since the cardholder's account with the 3DS Requestor had a change or account reset. 2 characters. Values accepted: 01, 02, 03, 04, 05	--	
nbPurchaseAccount	nbPurchaseAccount element of AReq; Number of purchases with this cardholder account during the previous six months. maximum 4 characters	--	
provisionAttemptsDay	provisionAttemptsDay element of AReq; Number of Add Card attempts in the last 24 hours. Maximum 3 characters	--	

AdditionalParams	Table 12		
txnActivityDay	txnActivityDay element of AReq; Number of transactions (successful and abandoned) for this cardholder account with the 3DS Requestor across all payment accounts in the previous 24 hours. Maximum 3 characters	--	
txnActivityYear	txnActivityYear element of AReq; Number of transactions (successful and abandoned) for this cardholder account with the 3DS Requestor across all payment accounts in the previous year. Maximum 3 characters	--	
paymentAccAge	paymentAccAge element of AReq; Date that the payment account was enrolled in the cardholder's account with the 3DS Requestor. 8 characters. Format accepted: YYYYMMDD	--	
paymentAccInd	paymentAccInd element of AReq; Indicates the length of time that the payment account was enrolled in the cardholder's account with the 3DS Requestor. 2 characters. Value accepted: 01, 02, 03, 04, 05	--	
shipAddressUsage	shipAddressUsage element of AReq; Date when the shipping address used for this transaction was first used with the 3DS Requestor. 8 characters. Format accepted: YYYYMMDD	--	
shipAddressUsageInd	shipAddressUsageInd element of AReq; Indicates when the shipping address used for this transaction was first used with the 3DS Requestor. 2 characters. Value accepted: 01, 02, 03, 04	--	
shipNameIndicator	shipNameIndicator element of AReq; Indicates if the Cardholder Name on the account is identical to the shipping Name used for this transaction. 2 characters. Value accepted: 01, 02	--	
suspiciousAccActivity	suspiciousAccActivity element of AReq; Indicates whether the 3DS Requestor has experienced suspicious activity previous fraud) on the cardholder account. 2 characters. Value accepted: 01, 02	--	

AdditionalParams	Table 12		
threeDSCompInd	threeDSCompInd element of AReq; Indicates whether the 3DS Method successfully completed. 1 character. Value accepted: U, Y, N	--	
threeDSRequestorAuthenticationInd	threeDSRequestorAuthenticationInd element of AReq; Indicates the type of Authentication request. 2 characters. Value accepted: 01, 02, 03, 04, 05, 06	--	
threeDSRequestorChallengeInd	threeDSRequestorChallengeInd element of AReq; Indicates whether a challenge is requested for this transaction. 2 characters. Value accepted: 01, 02, 03, 04	--	
threeRIInd	threeRIInd element of AReq; Indicates the type of 3RI request. 2 characters. Value accepted: 01, 02, 03, 04, 05	--	
addrMatch	addrMatch element of AReq; Indicates whether the Cardholder Shipping Address and Cardholder Billing Address are the same. 1 character. Value accepted: Y, N	--	
acctID	acctID element of AReq; Additional information about the account optionally provided by the 3DS Requestor. Maximum 64 characters	--	
email	email element of AReq; The email address associated with the account that is either entered by the Cardholder, or is on file with the 3DS Requestor. Maximum 254 characters	--	
homePhone.cc	homePhone element of AReq; The country code of home phone number provided by the Cardholder. 1-3 characters	--	
homePhone.subscriber	homePhone element of AReq; The subscriber of the home phone number provided by the Cardholder. maximum 15 characters	--	
mobilePhone.cc	mobilePhone element of AReq; The country code of mobile phone number provided by the Cardholder. 1-3 characters	--	

AdditionalParams	Table 12		
mobilePhone.subscriber	mobilePhone element of AReq; The subscriber of the mobile phone number provided by the Cardholder. maximum 15 characters	--	
workPhone.cc	workPhone element of AReq; The country code of work phone number provided by the Cardholder. 1-3 characters	--	
workPhone.subscriber	workPhone element of AReq; The subscriber of the work phone number provided by the Cardholder. maximum 15 characters	--	

Pre Authentication Response

A response message should be sent back by the remote authentication system to decide on the continuation of the authentication process.

Table 13 - PreAuthResp

PreAuthResp	Table 13		
Attribute	**Description**	**Usage**	**Sample Value**
Code	Response code: 0 - The authentication will be exempted. The authentication will not be displayed and the appropriate response will be returned. 1 - The transaction is not exempt. The authentication page will be displayed. 2 - There was an error but ActiveAccess will display the authentication page and let the authentication continue. ActiveAccess will not cancel the authentication. 3 - The transaction is deemed to be high risk, ActiveAccess will decline the transaction.	Required	2

PreAuthResp	Table 13		
AuthType	The comma separated list of decided authTypes by risk engine integration: 1- Password 2- SMS 3- OTP device 4- Virtual OTP device 5- CAP/DPA 6- Verify by Voice 7- USS 8- Q&A 9- OLB 10- CR 11- BIO 12- PKI 13- TTP 14- Email 15- OOB	Optional	2, 14
ErrorMessage	A descriptive message that identifies the category of the error	Optional	No card(s) found
ErrorDetail	A more detailed description of the error	Optional	No card(s) matching the request were found

Initiate Authentication

The Initiate Authentication step is optional and depends upon the type of authentication device being used. Once the registration status of the cardholder has been determined, ActiveAccess may initiate the authentication process by sending a request to the issuer's remote system. This step may be used for the first, and subsequent, generate challenge requests.

This step will commonly be used to initiate out of band authentication such as SMS, Question and Answer, Challenge and Response and Email.

> **Warning**
>
> This messaging is generally used only for out of band authentication and may be initiated either automatically by the system or manually, such as when a cardholder clicks on a "Send SMS" button on the page.

Initiate Authentication Request

Table 14 - InitAuthReq

InitAuthReq	Table 14		
Attribute	**Description**	**Usage**	**Sample Value**
Card	Where a value for Card.ID or Context_Blob was returned in the VerifyReg response, this value should be assigned to the Card.ID attribute. Otherwise, attributes of the Card may include **Number, Name** and **Type** as described in the VerifyReg request. Refer to *Table 3 - Card*.	Required. Either ID or Number and Type should be presented	card=[id=4564260131003313, number=4564-26XX-XXXX-3313, type=VbV, cardName=< null >, Context_Blob=595],
Transaction	Where messaging commences after the ActiveAccess system receives the PAReq, additional transaction, cardholder and merchant information is available. This information may be additionally sent to the issuer system for analysis and fraud detection purposes. Where required, the following data fields may be sent to the issuer's system in any of the request messages. Refer to *Table 4 - Transaction*.	Optional	transaction=[xid=MDAwMDAwMDAwM, purchaseAmount=12365, purchaseCurrency=840, purchaseDate=[orig_eon=< null >, orig_year=2016, orig_month=11, orig_day=3, orig_hour=10, orig_minute=23, orig_second=19, orig_fracSeconds=0.000, orig_timezone=210, eon=< null >, year=2016, month=11, day=3, timezone=210, hour=10, minute=23, second=19, fractionalSecond=0.000],

InitAuthReq	Table 14		
SMS	**Template** - The SMS message to be sent to the cardholder populated with Transaction.MerchantName, Transaction.PurchaseAmount and Transaction.PurchaseCurrency in the format that is required by SMS Gateway to send to customer mobile. template = "Sample message here. Your OTP is {0}". **Notes:** 1. The {0} is the placeholder where CAAS injects the actual 6-digit OTP. 2. {0} can be anywhere in the template – the above is just an example. 3. The length of the text can be up to 160 chars (note, the {0} placeholder will expand from 4 characters to 6 characters, so free text is effectively 154 characters.)	Conditional, where the authentication channel is SMS. Up to 154 characters.	Your OTP is :{0} \r\n merName: Test Merchant, purchaseAmount: 123.65

Email	Contains **Content**, **Subject** and the **Content-Type** of the email. **Content** (Required) - The content of the email to be sent to the cardholder, which can be populated with Transaction.MerchantName, Transaction.PurchaseAmount, Transaction.PurchaseCurrency, and any other information in the format that is configured by the bank to send to the customer's email address. **Notes:** 1. The {0} is the placeholder where CAAS injects the actual 6-digit OTP. 2. {0} can be anywhere in the template – the above is just an example. 3. The length of the text can be up to 160 chars (note, the {0} placeholder will expand from 4 characters to 6 characters, so free text is effectively 154 characters.) **Subject** (Required) - The subject of the email to be sent to the cardholder, which can be populated with *Issuer Name,* to send to the customer's email address. **Content-Type** (Required) - The content type of the email to be sent to the cardholder. This can be TEXT/PLAIN or TEXT/HTML.	Conditional. Content up to 1024 characters. Subject up to 998 characters. Content-Type up to 25 characters.

InitAuthReq	Table 14		
OobInfo	**Template** - The message to be sent to the OOB application populated with Transaction. MerchantName, Transaction.PurchaseAmount and Transaction.PurchaseCurrency in the format that is required by OOB adapter to send to OOB. template = ": "$ThreeDSServerTransID", "purchaseAmount": "$PurchaseAmount", "purchaseCurrency": "$PurchaseCurrency", "purchaseExponent": "$PurchaseExponent", "purchaseDate":"$PurchaseDate", "messageCategory": "$MessageCategory", "deviceChannel": "$DeviceChannel", "acctNumber": "$AcctNumber", "merchantName": "$MerchantName", "cardHolderInfo": { "cardholderName": "$CardholderName", "email": "$Email", "homePhone": { "cc": "$HomePhone_cc", "subscriber": "$HomePhone_subscriber" }, "mobilePhone": { "cc": "$MobilePhone_cc", "subscriber": "$MobilePhone_subscriber" }, "shipAddrCity": "$ShipAddrCity", "shipAddrCountry": "$ShipAddrCountry", "shipAddrLine1": "$ShipAddrLine1", "shipAddrLine2": "$ShipAddrLine2", "shipAddrLine3": "$ShipAddrLine3", "shipAddrPostCode": "$ShipAddrPostCode", "shipAddrState": "$ShipAddrState", "workPhone": { "cc": "$WorkPhone_cc", "subscriber": "$WorkPhone_subscriber" } } }	Conditional. Where the authentication channel is any of OOB 6 - Verify by Voice 7 - USS 9 - OLB 11 - BIO 13 - TTP 15 - OOB. Up to 4000 characters.	{ "threeDSServerTransID": "$ThreeDSServerTransID", "purchaseAmount": "123", "purchaseCurrency": "840", "purchaseExponent": "2", "purchaseDate":"20201216041928", "messageCategory": "01", "deviceChannel": "01", "acctNumber": "4123XXXXXXXX45", "merchantName": "Tet Merchant", "cardHolderInfo": { "cardholderName": "John", "email": "email@example.com", "homePhone": { "cc": "1", "subscriber": "530123112345" }, "mobilePhone": { "cc": "55", "subscriber": "23451443212" }, "shipAddrCity": "$ShipAddrCity", "shipAddrCountry": "$ShipAddrCountry", "shipAddrLine1": "$ShipAddrLine1", "shipAddrLine2": "$ShipAddrLine2", "shipAddrLine3": "$ShipAddrLine3", "shipAddrPostCode": "$ShipAddrPostCode", "shipAddrState": "$ShipAddrState", "workPhone": { "cc": "$WorkPhone_cc", "subscriber": "$WorkPhone_subscriber" } } }

InitAuthReq	Table 14		
callBack	**URL** - The URL that will receive the notification for the completion of OOB authentication.	Conditional. Where the authentication channel is any of OOB 6 - Verify by Voice 7 - USS 9 - OLB 11 - BIO 13 - TTP 15 - OOB.	http://acs.local:8080/acs/notifier/c26fe6e0-e8c6-45da-a488-09b9b50b82a6

InitAuthReq	Table 14		
AuthType	Authentication Type that cardholder requests to (re)initiate the one-time passcode for authentication: 2 - SMS 6 - Verify by Voice 7 - USS 10 - CR 11 - BIO 14 - Email 15 - OOB	Conditional. Up to 2 characters.	2

InitAuthReq	Table 14		
IV	If an encryption KeyStore has been defined for the issuer or group of issuers, critical card data must be encrypted by ActiveAccess using DESede/CBC/PKCS5Padding mode. The CBC encryption mode requires an Initialisation Vector (IV), which includes 8 random bytes, as an input parameter for encryption and decryption. The IV should be sent to the CAAS server to be used at decryption time. To do this, the IV which was used for encryption, must be encrypted in DESede/ECB/PKCS5Padding mode, using the same key, then HEX encoded and set as IV in the request.	Optional. If present, it means that ActiveAccess has generated an IV parameter and critical card information has been encrypted using the CBC mode and the generated IV, otherwise ECB or plain mode has been used instead. 8 or 16 characters in length.	8F51F71064DB2B65

Initiate Authentication Response

A response message should be sent back by the remote authentication system to indicate the status of sending an SMS or Email, or otherwise return AuthData for the authentication initiation.

Table 15 - InitAuthResp

InitAuthResp	Table 15		
Attribute	**Description**	**Usage**	**Sample Value**

InitAuthResp	Table 15		
Code	Response code: 0 - the request was successful 1 - there was an error and ActiveAccess should send the request again 2 - there was an error and ActiveAccess should cancel the authentication.	Required	2
ErrorMessage	A descriptive message that identifies the category of the error	Required	No card(s) found
ErrorDetail	A more detailed description of the error	Required	No card(s) matching the request were found
AuthData	Attributes of Data: **Name (required)** - the name of the AuthData parameter to be used for the data collection/disaply on the page, **AuthType (conditional)** - the AuthType of AuthData which will be used, **Format (optional)** - the regular expression for verifying the value collected from the page, **Mask (optional)** - true - input on the page will be masked. false - input on the page will be in plaintext, Refer to *Table 8 - Data*.	Conditional. Required if AuthTypeSup / AuthType is 2, 10, or 14. Recommended to mask the critical data such as mobile number.	authData=[data={[value=abababa, error=<null>, name=challenge, authType=10, format=\w+, mask=<null>, confirm=<null>],[value=<null>, error=<null>, name=response, authType=10, format=<null>, mask=<null>, confirm=<null>], [value=+421XXXXX1234, error=<null>, name=mobileNo, authType=2, format=<null>, mask=<null>, confirm=<null>]}],

Verify Authentication

Where the remote system determines that authentication is required and after an authentication has been initiated, the cardholder should be presented with an appropriate page. In many circumstances, this page will request the cardholder to enter their authentication credentials, such as a password or a one-time password. However, in some circumstances, the screen presented may ask the cardholder to press a button after having completed their out of band authentication.

When a cardholder enters their password, ActiveAccess will format the details of the authentication request and send it to the remote system for verification. The response provided will determine the authentication status of the transaction, with ActiveAccess formatting the 3-D Secure payer authentication response message to be returned to the merchant's MPI.

Verify Authentication Request

Table 16 - VerifyAuthReq

VerifyAuthReq	Table 16	
Attribute	**Description**	**U**
Card	Where a value for Card.ID or Card.Context_Blob has returned in the VerifyReg response, this value should be assigned to the Card.ID attribute. Otherwise, attributes of the Card may include **Number, Name** and **Type** as described in the VerifyReg request. Refer to *Table 3 - Card*.	F T
Token	The authentication number or password entered by the cardholder If an encryption KeyStore has been defined for the issuer or group of issuers, the token will be encrypted by ActiveAccess using DESede/CBC/PKCS5Padding mode and IV, then HEX encoded and included in the message. CAAS server will need to decrypt this field using DESede/CBC/PKCS5Padding mode and the request's IV before using it in the process.	C N a c V t b C t

VerifyAuthReq	Table 16	
AuthData	Attributes of Data: Name (required) - the name of the AuthData parameter to be used for data collection on the page, AuthType (conditional) - the AuthType of AuthData which has an error, Format (optional) - the regular expression for verifying the value collected from the page, **Mask (optional)** - true - input on the page will be masked false - input on the page will be in plaintext, **Confirm (optional)** - true - an additional input field will be added to the page for confirmation f the AuthData, and ACS will check that the two inputs match false - no confirmation input field will be displayed on the page Refer to *Table 8 - Data*.	C /
Transaction	Additional transaction information may include transaction, cardholder and merchant information such as **XID, PurchaseDate**, **PurchaseAmount**, **PurchaseCurrency**, **PurchaseDesc**, **MerchantID**, **AcqBIN**, **MerchantName**, **MerchantURL**, **MerchantCountry**, **CardExpiry** and **CardholderIP** as described in the Initiate Authentication request section. Refer to *Table 4 - Transaction*.	C

VerifyAuthReq	Table 16	
IV	If an encryption KeyStore has been defined for the issuer or group of issuers, critical card data must be encrypted by ActiveAccess using DESede/CBC/PKCS5Padding mode. The CBC encryption mode requires an Initialisation Vector (IV), which includes 8 random bytes, as an input parameter for encryption and decryption. The IV should be sent to the CAAS server to be used at decryption time. To do this, the IV which was used for encryption, must be encrypted in DESede/ECB/PKCS5Padding mode, using the same key, then HEX encoded and set as IV in the request.	C A p i t c b
HeaderParams	Attributes of Param: **Value** (required) **Key** (required) **Cookie** (optional)	C
ExtensionParams	Attributes of Param: **Value** (required) **Key** (required)	C

Verify Authentication Response

Table 17 - VerifyAuth

A response message should be sent back by the remote authentication system to indicate the success, or otherwise of the authentication verification.

VerifyAuthResp	Table 17		
Attribute	**Description**	**Usage**	**Sample Value**
Code	Response code: 0 - the authentication was successful 1 - the authentication token was incorrect 2 - an error occurred and another attempt should be made 3 - the status of the card is locked 4 - an error occurred and no further attempts should be made 5 - the authentication failed, transaction should end.	Required	3

VerifyAuthResp	Table 17		
AuthData	Attributes of Data: **Name** (required) - the name of the AuthData parameter to be used for data collection on the page **AuthType** (conditional) - the AuthType of AuthData which will be displayed on the authentication page **Format** (optional) - the regular expression for verifying the value collected from the page **Mask** (optional) - true - input on the page will be masked false - input on the page will be in plaintext **Confirm** (optional) - true - an additional input field will be added to the page for confirmation of the AuthData, and ACS will check that the two inputs match false - no confirmation input field will be displayed on the page Refer to *Table 8 - Data*.	Optional. When an error occurs, appropriate content can be returned. Otherwise, null will be returned.	authData=[data={[value=abababa, error=<null>, name=challenge, authType=10, format=<null>, mask=<null>, confirm=<null>],[value=1111, error=<null>, name=response, authType=10, format=<null>, mask=<null>, confirm=<null>]}]
ErrorMessage	A descriptive message that identifies the category of the error	Required	Card is locked
ErrorDetail	A more detailed description of the error	Required	The status of card is locked due to multiple unsuccessful login tries.
HeaderParams	Attributes of Param: Value (required) Key (required) Cookie (optional)	Optional	headerParams=[param={[value=value1, key=key1, cookie=true],...}],

VerifyAuthResp	Table 17		
ExtensionParams	Attributes of Param: Value (required) Key (required).	Optional	extensionParams=[param={[value=value1, key=key1],...}],

Verify Identity

Verify Identity data is used in ADS or the Forgot password process to primarily verify the identity of the cardholder before changing/setting authentication data.

A request message should be sent to CAAS with the user identity data and to have CAAS verify the data.

Verify Identity Request

Table 18 – VerifyIdentityReq

VerifyIdentityReq	Table 18		
Attribute	**Description**	**Usage**	**Sample Value**
Purpose	An attribute which indicates if Identity data are for the ADS or Forgot password process. 1 = reset password 2 = ADS	Required	1
IdentityData	Attributes of Data: **Name** (required) - the name of the IdentityData parameter to be used for data collection on the page **AuthType** (conditional) - empty value **Format** (optional) - empty value **Mask** (optional) - empty value **Confirm** (optional) - empty value Refer to **Table 8 - Data**.	Required	identityData=[data={[value=User1, error=<(null)>, name=cname, authType=<(null)>, format=<(null)>,, mask=<<(null)>,, confirm=<(null)>,], [value=123456, error=<(null)>,, name=pin, authType=<(null)>,, format=<(null)>,, mask=<(null)>, confirm=<(null)>,]}]
Transaction	Additional transaction information may include transaction, cardholder and merchant information such as **XID**, **PurchaseDate**, **PurchaseAmount**, **PurchaseCurrency**, **PurchaseDesc**, **MerchantID**, **AcqBIN**, **MerchantName**, **MerchantURL**, **MerchantCountry**, **CardExpiry** and **CardholderIP** Refer to *Table 4 - Transaction* for details.	Optional	

VerifyIdentityReq	Table 18		
IV	If an encryption KeyStore has been defined for the issuer or group of issuers, critical card data must be encrypted by ActiveAccess using DESede/CBC/PKCS5Padding mode. The CBC encryption mode requires an Initialisation Vector (IV), which includes 8 random bytes, as an input parameter for encryption and decryption. The IV should be sent to the CAAS server to be used at decryption time. To do this, the IV which was used for encryption, must be encrypted in DESede/ECB/PKCS5Padding mode, using the same key, then HEX encoded and set as IV in the request.	Optional. If present, it means that ActiveAccess has generated an IV parameter and critical card information has been encrypted using the CBC mode and the generated IV, otherwise ECB or plain mode has been used instead.	8F51F71064DB2B65
Card	Where a value for Card.ID or Card. Context_Blob has been returned in the VerifyReg response, this value should be assigned to the Card.ID attribute. Otherwise, attributes of the Card may include **Number, Name** and **Type** as described in the VerifyReg request. Refer to *Table 3 - Card* for details.	Required. Either ID or Number and Type should be presented.	card=[id=4564260131003313, number=4564-26XX-XXXX-3313, type=VbV, cardName=<(null)>, Context_Blob=595]

Verify Identity Response

A response message should be sent back to ActiveAccess to inform whether user identity has been verified or not and to return the reason in case of identity failure. In addition, it returns failed identity items to be highlighted in the page. In the case of a successful response, it returns AuthData to be asked for a subsequent authentication process.

Table 19 - VerifyIdentityResp

VerifyIdentityResp	Table 19		
Attribute	**Description**	**Usage**	**Sample Value**
Code	Response code: 0 - the authentication was successful, 1 - the authentication token was incorrect, 2 - an error occurred and another attempt should be made, 3 - the status of the card is locked, 4 - an error occurred and no further attempts should be made.	Required	3

VerifyIdentityResp	Table 19		
IdentityData	Attributes of Data: Name (required) - The name of the IdentityData parameter to be used for data collection on the page, AuthType (conditional) - empty value, Format (optional) - the regular expression for verifying the value collected from the page, **Mask (optional)** - true - input on the page will be masked false - input on the page will be in plaintext **Confirm (optional)** - true - an additional input field will be added to the page for confirmation of the AuthData, and ACS will check that the two inputs match false - no confirmation input field will be displayed on the page Refer to Table 8 - Data.	Optional. When an error occurs, appropriate content can be returned. Otherwise, null will be returned.	identityData=[data={[value=administrator11, error=[code=1, message=value mismatch, detail=value mismatch], name=cname, authType=<(null)>, format=<(null)>,, mask=<(null)>,, confirm=<(null)>],[value=123456, error=<(null)>, name=pin, authType=<(null)>, format=<(null)>, mask=<(null)>, confirm=<(null)>]}]

VerifyIdentityResp	Table 19		
AuthData	Attributes of Data: Name (required) - the name of the AuthData parameter which will be registered or reset for the card, AuthType (conditional) - the AuthType of AuthData which will be registered or reset for the card, Format (optional) - the regular expression for verifying the value collected from the page, **Mask (optional)** - true - input on the page will be masked false - input on the page will be in plaintext, **Confirm (optional)** - true - an additional input field will be added to the page for confirmation of the AuthData, and ACS will check that the two inputs match false - no confirmation input field will be displayed on the page Refer to *Table 8 - Data*.	Required. If VerifyIdentityReq.purpose=1, reset AuthData will be returned. If VerifyIdentityReq.purpose=2, a list of all AuthData will be returned for the cardholder to choose from and register with.	authData=[data={[value=<(null)>, error=<(null)>, name=password, authType=1, format=<(null)>, mask=true, confirm=true],[value=<(null)>, error=<(null)>, name=mobileNo, authType=2, format=<(null)>, mask=<(null)>, confirm=true], [value=<(null)>, error=<(null)>, name=token, authType=2, format=<(null)>, mask=<(null)>, confirm=<(null)>] … }]
ErrorMessage	A descriptive message that identifies the category of the error	Optional	Card is locked

VerifyIdentityResp	Table 19		
ErrorDetail	A more detailed description of the error	Optional	The status of card is locked due to multiple unsuccessful login attempts.

Register

A request message should be sent to CAAS to set Authentication data for subsequent authentication.

Register Request

Table 20 – RegisterReq

RegisterReq	Table 20	
Attribute	**Description**	**Usag**
RegisterData	Attributes of Data: Name (required) - the name of the collected RegisterData from the page, AuthType (conditional) - the AuthType of the collected RegisterData from the page, Format (optional) - empty value, Mask (optional) - empty value, Confirm (optional) - empty value. Refer to *Table 8 - Data*.	Requ
Card	Where a value for Card.ID or Card. Context_Blob has returned in the VerifyReg response, this value should be assigned to the Card.ID attribute. Otherwise, attributes of the Card may include **Number, Name** and **Type** as described in the VerifyReg request. Refer to *Table 3 - Card*.	Requ
Transaction	Additional transaction information may include transaction, cardholder and merchant information such as **XID, PurchaseDate**, **PurchaseAmount**, **PurchaseCurrency**, **PurchaseDesc**, **MerchantID**, **AcqBIN**, **MerchantName**, **MerchantURL**, **MerchantCountry**, **CardExpiry** and **CardholderIP.** Refer to *Table 4 - Transaction*.	Optio

RegisterReq	Table 20	
IV	If an encryption KeyStore has been defined for the issuer or group of issuers, critical card data must be encrypted by ActiveAccess using DESede/CBC/PKCS5Padding mode. The CBC encryption mode requires an Initialisation Vector (IV), which includes 8 random bytes, as an input parameter for encryption and decryption. The IV should be sent to the CAAS server to be used at decryption time. To do this, the IV which was used for encryption, must be encrypted in DESede/ECB/PKCS5Padding mode, using the same key, then HEX encoded and set as IV in the request.	Optio prese mear Activ has gener IV pa and c card inforr has b encry using CBC r and t gener other ECB c mode been instea

Register Response

Table 21 – RegisterResp

RegisterResp	Table 21		
Attribute	**Description**	**Usage**	**Sample Value**
Code	Response code: 0 - the authentication was successful, 1 - the authentication token was incorrect, 2 - an error occurred and another attempt should be made, 3 - the status of the card is locked, 4 - an error occurred and no further attempts should be made.	Required	3
RegisterData	Attributes of Data: Name (required) - the name of the RegisterData parameter, **Format (optional)** - the regular expression of RegisterData for verifying the value which is collected from the page, **Mask (optional)** - true - input on the page will be masked false - input on the page will be in plaintext, **Confirm (optional)** - true - an additional input field will be added to the page for confirmation of the AuthData, and ACS will check that the two inputs match false - no confirmation input field will be displayed on the page.	Optional. If an error occurs, registerData would be sent back to ACS with an appropriate error message. Refer to *Table 8 - Data*.	registerData=[data={ [value=<(null)>, error=[code=1, message=invalid, detail=invalid], name=password, authType=1, format=<(null)>, mask=<(null)>, confirm=<(null)>]}]

RegisterResp	Table 21		
ErrorMessage	A descriptive message that identifies the category of the error	Optional	
ErrorDetail	A more detailed description of the error	Optional	

Reset Password

Reset Password Request

A request message should be sent to CAAS for ResetPasswordData and have data set for further use.

Table 22 - ResetPasswordReq

ResetPasswordReq	Table 22
Attribute	**Description**
ResetPasswordData	Attributes of Data: Name (required) - the name of the ResetPasswordData parameter, collected from the page, AuthType (conditional) - the AuthType of ResetPasswordData, collected from the page, Format (optional) - empty value, Mask (optional) - empty value, Confirm (optional) - empty value Refer to *Table 21 – ResetPasswordResp*.
Card	Where a value for Card.ID or Card. Context_Blob has been returned in the VerifyReg response, this value should be assigned to the Card.ID attribute. Otherwise, attributes of the Card may include **Number, Name** and **Type** as described in the VerifyReg request. Refer to *Table 3 - Card*.
Transaction	Additional transaction information may include transaction, cardholder and merchant information such as **XID, PurchaseDate**, **PurchaseAmount**, **PurchaseCurrency**, **PurchaseDesc**, **MerchantID**, **AcqBIN**, **MerchantName**, **MerchantURL**, **MerchantCountry**, **CardExpiry** and **CardholderIP**. Refer to *Table 4 - Transaction*.

ResetPasswordReq	Table 22
IV	If an encryption KeyStore has been defined for the issuer or group of issuers, critical card data must be encrypted by ActiveAccess using DESede/CBC/PKCS5Padding mode. The CBC encryption mode requires an Initialisation Vector (IV), which includes 8 random bytes, as an input parameter for encryption and decryption. The IV should be sent to the CAAS server to be used at decryption time. To do this, the IV which was used for encryption, must be encrypted in DESede/ECB/PKCS5Padding mode, using the same key, then HEX encoded and set as IV in the request.

Reset Password Response

A response message should be sent back to ActiveAccess to indicate the result of the reset password process in CAAS.

Table 23 – ResetPasswordResp

ResetPasswordResp	Table 23		
Attribute	**Description**	**Usage**	**Sample Value**
Code	Response code: 0 - the authentication was successful, 1 - the authentication token was incorrect, 2 - an error occurred and another attempt should be made, 3 - the status of the card is locked, 4 - an error occurred and no further attempts should be made.	Required	3

ResetPasswordResp	Table 23		
ResetPasswordData	Attributes of Data: Name (required) - the name of the ResetPasswordData parameter, **AuthType (conditional)** - the AuthType of ResetPasswordData, **Format (optional)** - the regular expression for verifying the value collected from the page, **Mask (optional)** - true - input on the page will be masked true - input on the page will be masked false - input on the page will be in plaintext, **Confirm (optional)** - true - an additional input field will be added to the page for confirmation of the AuthData, and ACS will check that the two inputs match false - no confirmation input field will be displayed on the page Refer to **Table 8 - Data**.	Optional. When an error occurs, appropriate content can be returned.	resetPasswordData= [data={[value=<(null)>, error=[code=1, message=invalid, detail=invalid], name=password, authType=1, format=<(null)>, mask=<(null)>, confirm=<(null)>]}]
ErrorMessage	A descriptive message that identifies the category of the error	Optional	
ErrorDetail	A more detailed description of the error	Optional	

Ping

The ping request is used to determine the responsiveness and availability of the server. Simply send a ping request to the server to check if the service is up and operational or not.

Ping Request

Ping has no request parameter.

Ping Response

Ping has no response. Successful return of the operation invocation without any exception means the service is up and running.

Messaging Requirements

Securing Message Channel

Communication security must be ensured by using SSL with server and client authentication.

Critical Card Data Encryption and Decryption

The key, which is used for encrypting/decrypting the critical card data, must be a 112 or 168 bit DESede key. A KeyStore with the following details should be prepared for the encryption key that is to be uploaded, through MIA, for the specified issuer or group of issuers:

KeyStore type/format: JCEKS

KeyStore provider: SunJCE

Key algorithm: DESede

Key size: 112 or 168 bit

Key name: can be any

No of keys in the KeyStore: Only one key must be populated in the KeyStore

Such KeyStores can be easily created through the Java keytool utility using the following command:

```
keytool -genseckey -alias enckey168 -keypass 123456 -keyalg DESede -keysize 168 -
keystore enc-key.JKS -storepass 123456 -storetype JCEKS
```

If IV is set for the request, the CAAS server needs to get the IV by HEX decoding and decrypting the VerifyRegReq.IV / InitAuthReq.IV / VerifyAuthReq.IV using the encryption key in DESede/ECB/

PKCS5Padding mode, before decrypting the critical card data in DESede/CBC/PKCS5Padding mode using the obtained IV from the request.

Calling Convention

Requests will be sent using SOAP on HTTPS.

Remote System Integration WSDL

> **Important**
>
> It is important to ensure messages conform to the requirements of the remote system integration API by validating them against the WSDL and XSD schema.
>
> The Remote System Integration WSDL and XSD schema can be found in the ActiveAccess installation package in the following path:
>
> `ActiveAccess/files/acs.war/WEB-INF/lib/caas.client-*.jar`

Country and Currency Codes

ISO 4217 Currency Codes combined with ISO 3166 Country Codes

Note

Codes marked with an asterisk (*) are not used in ActiveAccess

ISO-3166 Country Codes				ISO 4217 Currency Codes				
Country	**Code**			**Country**	**Currency**	**Code**		**Minor unit**
	Alpha2	**Alpha3**	**Numeric 3**			Alphabetic	Numeric	
Afghanistan	AF	AFG	004	AFGHANISTAN	Afghani	AFN	971	2
Åland Islands	AX	ALA	248	ÅLAND ISLANDS	Euro	EUR	978	2
Albania	AL	ALB	008	ALBANIA	Lek	ALL	008	2
Algeria	DZ	DZA	012	ALGERIA	Algerian Dinar	DZD	012	2
American Samoa	AS	ASM	016	AMERICAN SAMOA	US Dollar	USD	840	2
Andorra	AD	AND	020	ANDORRA	Euro	EUR	978	2
Angola	AO	AGO	024	ANGOLA	Kwanza	AOA	973	2
Anguilla	AI	AIA	660	ANGUILLA	East Caribbean Dollar	XCD	951	2
Antarctica	AQ	ATA	010	ANTARCTICA*	No universal currency			

ISO-3166 Country Codes				ISO 4217 Currency Codes				
Antigua and Barbuda	AG	ATG	028	ANTIGUA AND BARBUDA	East Caribbean Dollar	XCD	951	2
Argentina	AR	ARG	032	ARGENTINA	Argentine Peso	ARS	032	2
Armenia	AM	ARM	051	ARMENIA	Armenian Dram	AMD	051	2
Aruba	AW	ABW	533	ARUBA	Aruban Florin	AWG	533	2
Australia	AU	AUS	036	AUSTRALIA	Australian Dollar	AUD	036	2
Austria	AT	AUT	040	AUSTRIA	Euro	EUR	978	2
Azerbaijan	AZ	AZE	031	AZERBAIJAN	Azerbaijan Manat	AZN	944	2
Bahamas (the)	BS	BHS	044	BAHAMAS (THE)	Bahamian Dollar	BSD	044	2
Bahrain	BH	BHR	048	BAHRAIN	Bahraini Dinar	BHD	048	3
Bangladesh	BD	BGD	050	BANGLADESH	Taka	BDT	050	2
Barbados	BB	BRB	052	BARBADOS	Barbados Dollar	BBD	052	2
Belarus	BY	BLR	112	BELARUS	Belarusian Ruble	BYN	933	2

ISO-3166 Country Codes				ISO 4217 Currency Codes				
				BELARUS	Belarusian Ruble	BYR	974	0
Belgium	BE	BEL	056	BELGIUM	Euro	EUR	978	2
Belize	BZ	BLZ	084	BELIZE	Belize Dollar	BZD	084	2
Benin	BJ	BEN	204	BENIN	CFA Franc BCEAO	XOF	952	0
Bermuda	BM	BMU	060	BERMUDA	Bermudian Dollar	BMD	060	2
Bhutan	BT	BTN	064	BHUTAN	Indian Rupee	INR	356	2
				BHUTAN	Ngultrum	BTN	064	2
Bolivia (Plurinational State of)	BO	BOL	068	BOLIVIA (PLURINATIONAL STATE OF)	Boliviano	BOB	068	2
				BOLIVIA (PLURINATIONAL STATE OF)	Mvdol	BOV	984	2
Bonaire, Sint Eustatius and Saba	BQ	BES	535	BONAIRE, SINT EUSTATIUS AND SABA	US Dollar	USD	840	2

ISO-3166 Country Codes				ISO 4217 Currency Codes				
Bosnia and Herzegovina	BA	BIH	070	BOSNIA AND HERZEGOVINA	Convertible Mark	BAM	977	2
Botswana	BW	BWA	072	BOTSWANA	Pula	BWP	072	2
Bouvet Island	BV	BVT	074	BOUVET ISLAND	Norwegian Krone	NOK	578	2
Brazil	BR	BRA	076	BRAZIL	Brazilian Real	BRL	986	2
British Indian Ocean Territory (the)	IO	IOT	086	BRITISH INDIAN OCEAN TERRITORY (THE)	US Dollar	USD	840	2
Brunei Darussalam	BN	BRN	096	BRUNEI DARUSSALAM	Brunei Dollar	BND	096	2
Bulgaria	BG	BGR	100	BULGARIA	Bulgarian Lev	BGN	975	2
Burkina Faso	BF	BFA	854	BURKINA FASO	CFA Franc BCEAO	XOF	952	0
Burundi	BI	BDI	108	BURUNDI	Burundi Franc	BIF	108	0
Cabo Verde	CV	CPV	132	CABO VERDE	Cabo Verde Escudo	CVE	132	2
Cambodia	KH	KHM	116	CAMBODIA	Riel	KHR	116	2

ISO-3166 Country Codes				ISO 4217 Currency Codes				
Cameroon	CM	CMR	120	CAMEROON	CFA Franc BEAC	XAF	950	0
Canada	CA	CAN	124	CANADA	Canadian Dollar	CAD	124	2
Cayman Islands (the)	KY	CYM	136	CAYMAN ISLANDS (THE)	Cayman Islands Dollar	KYD	136	2
Central African Republic (the)	CF	CAF	140	CENTRAL AFRICAN REPUBLIC (THE)	CFA Franc BEAC	XAF	950	0
Chad	TD	TCD	148	CHAD	CFA Franc BEAC	XAF	950	0
Chile	CL	CHL	152	CHILE	Chilean Peso	CLP	152	0
				CHILE	Unidad de Fomento	CLF	990	4
China	CN	CHN	156	CHINA	Yuan Renminbi	CNY	156	2
Christmas Island	CX	CXR	162	CHRISTMAS ISLAND	Australian Dollar	AUD	036	2
Cocos (Keeling) Islands (the)	CC	CCK	166	COCOS (KEELING) ISLANDS (THE)	Australian Dollar	AUD	036	2
Colombia	CO	COL	170	COLOMBIA	Colombian Peso	COP	170	2

ISO-3166 Country Codes				ISO 4217 Currency Codes				
				COLOMBIA	Unidad de Valor Real	COU	970	2
Comoros (the)	KM	COM	174	COMOROS (THE)	Comorian Franc	KMF	174	0
Congo (the Democratic Republic of the)	CD	COD	180	CONGO (THE DEMOCRATIC REPUBLIC OF THE)	Congolese Franc	CDF	976	2
Congo (the)	CG	COG	178	CONGO (THE)	CFA Franc BEAC	XAF	950	0
Cook Islands (the)	CK	COK	184	COOK ISLANDS (THE)	New Zealand Dollar	NZD	554	2
Costa Rica	CR	CRI	188	COSTA RICA	Costa Rican Colon	CRC	188	2
Côte d'Ivoire	CI	CIV	384	CÔTE D'IVOIRE	CFA Franc BCEAO	XOF	952	0
Croatia	HR	HRV	191	CROATIA	Kuna	HRK	191	2
Cuba	CU	CUB	192	CUBA	Cuban Peso	CUP	192	2
				CUBA	Peso Convertible	CUC	931	2
Curaçao	CW	CUW	531	CURAÇAO	Netherlands Antillean Guilder	ANG	532	2

ISO-3166 Country Codes				ISO 4217 Currency Codes				
Cyprus	CY	CYP	196	CYPRUS	Euro	EUR	978	2
CZECHIA	CZ	CZE	203	CZECHIA	Czech Koruna	CZK	203	2
Denmark	DK	DNK	208	DENMARK	Danish Krone	DKK	208	2
Djibouti	DJ	DJI	262	DJIBOUTI	Djibouti Franc	DJF	262	0
Dominica	DM	DMA	212	DOMINICA	East Caribbean Dollar	XCD	951	2
Dominican Republic (the)	DO	DOM	214	DOMINICAN REPUBLIC (THE)	Dominican Peso	DOP	214	2
Ecuador	EC	ECU	218	ECUADOR	US Dollar	USD	840	2
Egypt	EG	EGY	818	EGYPT	Egyptian Pound	EGP	818	2
El Salvador	SV	SLV	222	EL SALVADOR	El Salvador Colon	SVC	222	2
				EL SALVADOR	US Dollar	USD	840	2
Equatorial Guinea	GQ	GNQ	226	EQUATORIAL GUINEA	CFA Franc BEAC	XAF	950	0
Eritrea	ER	ERI	232	ERITREA	Nakfa	ERN	232	2

ISO-3166 Country Codes				ISO 4217 Currency Codes				
Estonia	EE	EST	233	ESTONIA	Euro	EUR	978	2
Ethiopia	ET	ETH	231	ETHIOPIA	Ethiopian Birr	ETB	230	2
				EUROPEAN UNION	Euro	EUR	978	2
Falkland Islands (the) [Malvinas]	FK	FLK	238	FALKLAND ISLANDS (THE) [MALVINAS]	Falkland Islands Pound	FKP	238	2
Faroe Islands (the)	FO	FRO	234	FAROE ISLANDS (THE)	Danish Krone	DKK	208	2
Fiji	FJ	FJI	242	FIJI	Fiji Dollar	FJD	242	2
Finland	FI	FIN	246	FINLAND	Euro	EUR	978	2
France	FR	FRA	250	FRANCE	Euro	EUR	978	2
French Guiana	GF	GUF	254	FRENCH GUIANA	Euro	EUR	978	2
French Polynesia	PF	PYF	258	FRENCH POLYNESIA	CFP Franc	XPF	953	0
French Southern Territories (the)	TF	ATF	260	FRENCH SOUTHERN TERRITORIES (THE)	Euro	EUR	978	2

ISO-3166 Country Codes				ISO 4217 Currency Codes				
Gabon	GA	GAB	266	GABON	CFA Franc BEAC	XAF	950	0
the Republic of the Gambia	GM	GMB	270	GAMBIA (THE)	Dalasi	GMD	270	2
Georgia	GE	GEO	268	GEORGIA	Lari	GEL	981	2
Germany	DE	DEU	276	GERMANY	Euro	EUR	978	2
Ghana	GH	GHA	288	GHANA	Ghana Cedi	GHS	936	2
Gibraltar	GI	GIB	292	GIBRALTAR	Gibraltar Pound	GIP	292	2
Greece	GR	GRC	300	GREECE	Euro	EUR	978	2
Greenland	GL	GRL	304	GREENLAND	Danish Krone	DKK	208	2
Grenada	GD	GRD	308	GRENADA	East Caribbean Dollar	XCD	951	2
Guadeloupe	GP	GLP	312	GUADELOUPE	Euro	EUR	978	2
Guam	GU	GUM	316	GUAM	US Dollar	USD	840	2
Guatemala	GT	GTM	320	GUATEMALA	Quetzal	GTQ	320	2

ISO-3166 Country Codes				ISO 4217 Currency Codes				
Guernsey	GG	GGY	831	GUERNSEY	Pound Sterling	GBP	826	2
Guinea	GN	GIN	324	GUINEA	Guinean Franc	GNF	324	0
Guinea-Bissau	GW	GNB	624	GUINEA-BISSAU	CFA Franc BCEAO	XOF	952	0
Guyana	GY	GUY	328	GUYANA	Guyana Dollar	GYD	328	2
Haiti	HT	HTI	332	HAITI	Gourde	HTG	332	2
				HAITI	US Dollar	USD	840	2
Heard Island and McDonald Islands	HM	HMD	334	HEARD ISLAND AND McDONALD ISLANDS	Australian Dollar	AUD	036	2
Holy See (the)	VA	VAT	336	HOLY SEE (THE)	Euro	EUR	978	2
Honduras	HN	HND	340	HONDURAS	Lempira	HNL	340	2
Hong Kong	HK	HKG	344	HONG KONG	Hong Kong Dollar	HKD	344	2
Hungary	HU	HUN	348	HUNGARY	Forint	HUF	348	2
Iceland	IS	ISL	352	ICELAND	Iceland Krona	ISK	352	0

ISO-3166 Country Codes				ISO 4217 Currency Codes				
India	IN	IND	356	INDIA	Indian Rupee	INR	356	2
Indonesia	ID	IDN	360	INDONESIA	Rupiah	IDR	360	2
				INTERNATIONAL MONETARY FUND (IMF) *	SDR (Special Drawing Right)	XDR	960	N.A.
Iran (Islamic Republic of)	IR	IRN	364	IRAN (ISLAMIC REPUBLIC OF)	Iranian Rial	IRR	364	2
Iraq	IQ	IRQ	368	IRAQ	Iraqi Dinar	IQD	368	3
Ireland	IE	IRL	372	IRELAND	Euro	EUR	978	2
Isle of Man	IM	IMN	833	ISLE OF MAN	Pound Sterling	GBP	826	2
Israel	IL	ISR	376	ISRAEL	New Israeli Sheqel	ILS	376	2
Italy	IT	ITA	380	ITALY	Euro	EUR	978	2
Jamaica	JM	JAM	388	JAMAICA	Jamaican Dollar	JMD	388	2
Japan	JP	JPN	392	JAPAN	Yen	JPY	392	0

ISO-3166 Country Codes				ISO 4217 Currency Codes				
Jersey	JE	JEY	832	JERSEY	Pound Sterling	GBP	826	2
Jordan	JO	JOR	400	JORDAN	Jordanian Dinar	JOD	400	3
Kazakhstan	KZ	KAZ	398	KAZAKHSTAN	Tenge	KZT	398	2
Kenya	KE	KEN	404	KENYA	Kenyan Shilling	KES	404	2
Kiribati	KI	KIR	296	KIRIBATI	Australian Dollar	AUD	036	2
Korea (the Democratic People's Republic of)	KP	PRK	408	KOREA (THE DEMOCRATIC PEOPLE'S REPUBLIC OF)	North Korean Won	KPW	408	2
Korea (the Republic of)	KR	KOR	410	KOREA (THE REPUBLIC OF)	Won	KRW	410	0
Kosovo [1]	QZ	QZZ	900	KOSOVO	Euro	EUR	978	2
Kuwait	KW	KWT	414	KUWAIT	Kuwaiti Dinar	KWD	414	3
Kyrgyzstan	KG	KGZ	417	KYRGYZSTAN	Som	KGS	417	2

ISO-3166 Country Codes				ISO 4217 Currency Codes				
Lao People's Democratic Republic (the)	LA	LAO	418	LAO PEOPLE'S DEMOCRATIC REPUBLIC (THE)	Lao Kip	LAK	418	2
Latvia	LV	LVA	428	LATVIA	Euro	EUR	978	2
Lebanon	LB	LBN	422	LEBANON	Lebanese Pound	LBP	422	2
Lesotho	LS	LSO	426	LESOTHO	Loti	LSL	426	2
				LESOTHO	Rand	ZAR	710	2
Liberia	LR	LBR	430	LIBERIA	Liberian Dollar	LRD	430	2
Libya	LY	LBY	434	LIBYA	Libyan Dinar	LYD	434	3
Liechtenstein	LI	LIE	438	LIECHTENSTEIN	Swiss Franc	CHF	756	2
Lithuania	LT	LTU	440	LITHUANIA	Euro	EUR	978	2
Luxembourg	LU	LUX	442	LUXEMBOURG	Euro	EUR	978	2
Macao	MO	MAC	446	MACAO	Pataca	MOP	446	2

ISO-3166 Country Codes				ISO 4217 Currency Codes				
Macedonia (the former Yugoslav Republic of)	MK	MKD	807	MACEDONIA (THE FORMER YUGOSLAV REPUBLIC OF)	Denar	MKD	807	2
Madagascar	MG	MDG	450	MADAGASCAR	Malagasy Ariary	MGA	969	2
Malawi	MW	MWI	454	MALAWI	Malawi Kwacha	MWK	454	2
Malaysia	MY	MYS	458	MALAYSIA	Malaysian Ringgit	MYR	458	2
Maldives	MV	MDV	462	MALDIVES	Rufiyaa	MVR	462	2
Mali	ML	MLI	466	MALI	CFA Franc BCEAO	XOF	952	0
Malta	MT	MLT	470	MALTA	Euro	EUR	978	2
Marshall Islands (the)	MH	MHL	584	MARSHALL ISLANDS (THE)	US Dollar	USD	840	2
Martinique	MQ	MTQ	474	MARTINIQUE	Euro	EUR	978	2
Mauritania	MR	MRT	478	MAURITANIA	Ouguiya	MRU	929	2
Mauritius	MU	MUS	480	MAURITIUS	Mauritius Rupee	MUR	480	2

ISO-3166 Country Codes				ISO 4217 Currency Codes				
Mayotte	YT	MYT	175	MAYOTTE	Euro	EUR	978	2
				MEMBER COUNTRIES OF THE AFRICAN DEVELOPMENT BANK GROUP*	ADB Unit of Account	XUA	965	N.A.
Mexico	MX	MEX	484	MEXICO	Mexican Peso	MXN	484	2
				MEXICO	Mexican Unidad de Inversion (UDI)	MXV	979	2
Micronesia (Federated States of)	FM	FSM	583	MICRONESIA (FEDERATED STATES OF)	US Dollar	USD	840	2
Moldova	MD	MDA	498	MOLDOVA (THE REPUBLIC OF)	Moldovan Leu	MDL	498	2
Monaco	MC	MCO	492	MONACO	Euro	EUR	978	2
Mongolia	MN	MNG	496	MONGOLIA	Tugrik	MNT	496	2
Montenegro	ME	MNE	499	MONTENEGRO	Euro	EUR	978	2
Montserrat	MS	MSR	500	MONTSERRAT	East Caribbean Dollar	XCD	951	2

ISO-3166 Country Codes				ISO 4217 Currency Codes				
Morocco	MA	MAR	504	MOROCCO	Moroccan Dirham	MAD	504	2
Mozambique	MZ	MOZ	508	MOZAMBIQUE	Mozambique Metical	MZN	943	2
Myanmar	MM	MMR	104	MYANMAR	Kyat	MMK	104	2
Namibia	NA	NAM	516	NAMIBIA	Namibia Dollar	NAD	516	2
				NAMIBIA	Rand	ZAR	710	2
Nauru	NR	NRU	520	NAURU	Australian Dollar	AUD	036	2
Nepal	NP	NPL	524	NEPAL	Nepalese Rupee	NPR	524	2
Netherlands (the)	NL	NLD	528	NETHERLANDS (THE)	Euro	EUR	978	2
New Caledonia	NC	NCL	540	NEW CALEDONIA	CFP Franc	XPF	953	0
New Zealand	NZ	NZL	554	NEW ZEALAND	New Zealand Dollar	NZD	554	2
Nicaragua	NI	NIC	558	NICARAGUA	Cordoba Oro	NIO	558	2
Niger (the)	NE	NER	562	NIGER (THE)	CFA Franc BCEAO	XOF	952	0

ISO-3166 Country Codes				ISO 4217 Currency Codes				
Nigeria	NG	NGA	566	NIGERIA	Naira	NGN	566	2
Niue	NU	NIU	570	NIUE	New Zealand Dollar	NZD	554	2
Norfolk Island	NF	NFK	574	NORFOLK ISLAND	Australian Dollar	AUD	036	2
Northern Mariana Islands (the)	MP	MNP	580	NORTHERN MARIANA ISLANDS (THE)	US Dollar	USD	840	2
Norway	NO	NOR	578	NORWAY	Norwegian Krone	NOK	578	2
Oman	OM	OMN	512	OMAN	Rial Omani	OMR	512	3
Pakistan	PK	PAK	586	PAKISTAN	Pakistan Rupee	PKR	586	2
Palau	PW	PLW	585	PALAU	US Dollar	USD	840	2
Palestine, State of	PS	PSE	275	PALESTINE, STATE OF*	No universal currency			
Panama	PA	PAN	591	PANAMA	Balboa	PAB	590	2
				PANAMA	US Dollar	USD	840	2
Papua New Guinea	PG	PNG	598	PAPUA NEW GUINEA	Kina	PGK	598	2

ISO-3166 Country Codes				ISO 4217 Currency Codes				
Paraguay	PY	PRY	600	PARAGUAY	Guarani	PYG	600	0
Peru	PE	PER	604	PERU	Sol	PEN	604	2
Philippines (the)	PH	PHL	608	PHILIPPINES (THE)	Philippine Piso	PHP	608	2
Pitcairn	PN	PCN	612	PITCAIRN	New Zealand Dollar	NZD	554	2
Poland	PL	POL	616	POLAND	Zloty	PLN	985	2
Portugal	PT	PRT	620	PORTUGAL	Euro	EUR	978	2
Puerto Rico	PR	PRI	630	PUERTO RICO	US Dollar	USD	840	2
Qatar	QA	QAT	634	QATAR	Qatari Rial	QAR	634	2
Réunion	RE	REU	638	RÉUNION	Euro	EUR	978	2
Romania	RO	ROU	642	ROMANIA	Romanian Leu	RON	946	2
Russian Federation (the)	RU	RUS	643	RUSSIAN FEDERATION (THE)	Russian Ruble	RUB	643	2
Rwanda	RW	RWA	646	RWANDA	Rwanda Franc	RWF	646	0

ISO-3166 Country Codes				ISO 4217 Currency Codes				
Saint Barthélemy	BL	BLM	652	SAINT BARTHÉLEMY	Euro	EUR	978	2
Saint Helena, Ascension and Tristan da Cunha	SH	SHN	654	SAINT HELENA, ASCENSION AND TRISTAN DA CUNHA	Saint Helena Pound	SHP	654	2
Saint Kitts and Nevis	KN	KNA	659	SAINT KITTS AND NEVIS	East Caribbean Dollar	XCD	951	2
Saint Lucia	LC	LCA	662	SAINT LUCIA	East Caribbean Dollar	XCD	951	2
Saint Martin (French part)	MF	MAF	663	SAINT MARTIN (FRENCH PART)	Euro	EUR	978	2
Saint Pierre and Miquelon	PM	SPM	666	SAINT PIERRE AND MIQUELON	Euro	EUR	978	2
Saint Vincent and the Grenadines	VC	VCT	670	SAINT VINCENT AND THE GRENADINES	East Caribbean Dollar	XCD	951	2
Samoa	WS	WSM	882	SAMOA	Tala	WST	882	2
San Marino	SM	SMR	674	SAN MARINO	Euro	EUR	978	2

ISO-3166 Country Codes				ISO 4217 Currency Codes				
Sao Tome and Principe	ST	STP	678	SAO TOME AND PRINCIPE	Dobra	STD	678	2
Sao Tome and Principe	ST	STN	930	SAO TOME AND PRINCIPE	Dobra	STN	930	2
Saudi Arabia	SA	SAU	682	SAUDI ARABIA	Saudi Riyal	SAR	682	2
Senegal	SN	SEN	686	SENEGAL	CFA Franc BCEAO	XOF	952	0
Serbia	RS	SRB	688	SERBIA	Serbian Dinar	RSD	941	2
Seychelles	SC	SYC	690	SEYCHELLES	Seychelles Rupee	SCR	690	2
Sierra Leone	SL	SLE	694	SIERRA LEONE	Leone	SLL	694	2
Singapore	SG	SGP	702	SINGAPORE	Singapore Dollar	SGD	702	2
Sint Maarten (Dutch part)	SX	SXM	534	SINT MAARTEN (DUTCH PART)	Netherlands Antillean Guilder	ANG	532	2
				SISTEMA UNITARIO DE COMPENSACION REGIONAL DE PAGOS "SUCRE"*	Sucre	XSU	994	N.A.

ISO-3166 Country Codes				ISO 4217 Currency Codes				
Slovakia	SK	SVK	703	SLOVAKIA	Euro	EUR	978	2
Slovenia	SI	SVN	705	SLOVENIA	Euro	EUR	978	2
Solomon Islands	SB	SLB	090	SOLOMON ISLANDS	Solomon Islands Dollar	SBD	090	2
Somalia	SO	SOM	706	SOMALIA	Somali Shilling	SOS	706	2
South Africa	ZA	ZAF	710	SOUTH AFRICA	Rand	ZAR	710	2
South Georgia and the South Sandwich Islands	GS	SGS	239	SOUTH GEORGIA AND THE SOUTH SANDWICH ISLANDS*	No universal currency			
South Sudan	SS	SSD	728	SOUTH SUDAN	South Sudanese Pound	SSP	728	2
Spain	ES	ESP	724	SPAIN	Euro	EUR	978	2
Sri Lanka	LK	LKA	144	SRI LANKA	Sri Lanka Rupee	LKR	144	2
Sudan (the)	SD	SDN	729	SUDAN (THE)	Sudanese Pound	SDG	938	2
Suriname	SR	SUR	740	SURINAME	Surinam Dollar	SRD	968	2

ISO-3166 Country Codes				ISO 4217 Currency Codes				
Svalbard and Jan Mayen	SJ	SJM	744	SVALBARD AND JAN MAYEN	Norwegian Krone	NOK	578	2
Swaziland	SZ	SWZ	748	SWAZILAND	Lilangeni	SZL	748	2
Sweden	SE	SWE	752	SWEDEN	Swedish Krona	SEK	752	2
Switzerland	CH	CHE	756	SWITZERLAND	Swiss Franc	CHF	756	2
				SWITZERLAND	WIR Euro	CHE	947	2
				SWITZERLAND	WIR Franc	CHW	948	2
Syrian Arab Republic	SY	SYR	760	SYRIAN ARAB REPUBLIC	Syrian Pound	SYP	760	2
Taiwan (Province of China)	TW	TWN	158	TAIWAN (PROVINCE OF CHINA)	New Taiwan Dollar	TWD	901	2
Tajikistan	TJ	TJK	762	TAJIKISTAN	Somoni	TJS	972	2
Tanzania, United Republic of	TZ	TZA	834	TANZANIA, UNITED REPUBLIC OF	Tanzanian Shilling	TZS	834	2
Thailand	TH	THA	764	THAILAND	Baht	THB	764	2

ISO-3166 Country Codes				ISO 4217 Currency Codes				
Timor-Leste	TL	TLS	626	TIMOR-LESTE	US Dollar	USD	840	2
Togo	TG	TGO	768	TOGO	CFA Franc BCEAO	XOF	952	0
Tokelau	TK	TKL	772	TOKELAU	New Zealand Dollar	NZD	554	2
Tonga	TO	TON	776	TONGA	Pa'anga	TOP	776	2
Trinidad and Tobago	TT	TTO	780	TRINIDAD AND TOBAGO	Trinidad and Tobago Dollar	TTD	780	2
Tunisia	TN	TUN	788	TUNISIA	Tunisian Dinar	TND	788	3
Turkey	TR	TUR	792	TURKEY	Turkish Lira	TRY	949	2
Turkmenistan	TM	TKM	795	TURKMENISTAN	Turkmenistan New Manat	TMT	934	2
Turks and Caicos Islands (the)	TC	TCA	796	TURKS AND CAICOS ISLANDS (THE)	US Dollar	USD	840	2
Tuvalu	TV	TUV	798	TUVALU	Australian Dollar	AUD	036	2
Uganda	UG	UGA	800	UGANDA	Uganda Shilling	UGX	800	0

ISO-3166 Country Codes				ISO 4217 Currency Codes				
Ukraine	UA	UKR	804	UKRAINE	Hryvnia	UAH	980	2
United Arab Emirates (the)	AE	ARE	784	UNITED ARAB EMIRATES (THE)	UAE Dirham	AED	784	2
United Kingdom of Great Britain and Northern Ireland (the)	GB	GBR	826	UNITED KINGDOM OF GREAT BRITAIN AND NORTHERN IRELAND (THE)	Pound Sterling	GBP	826	2
United States Minor Outlying Islands (the)	UM	UMI	581	UNITED STATES MINOR OUTLYING ISLANDS (THE)	US Dollar	USD	840	2
United States of America (the)	US	USA	840	UNITED STATES OF AMERICA (THE)	US Dollar	USD	840	2
				UNITED STATES OF AMERICA (THE)	US Dollar (Next day)	USN	997	2
Uruguay	UY	URY	858	URUGUAY	Peso Uruguayo	UYU	858	2
				URUGUAY	Uruguay Peso en Unidades Indexadas (UI)	UYI	940	0
Uzbekistan	UZ	UZB	860	UZBEKISTAN	Uzbekistan Sum	UZS	860	2

ISO-3166 Country Codes				ISO 4217 Currency Codes				
Vanuatu	VU	VUT	548	VANUATU	Vatu	VUV	548	0
Venezuela (Bolivarian Republic of)	VE	VEN	862	VENEZUELA (BOLIVARIAN REPUBLIC OF)	Bolívar Soberano	VES	928	2
Viet Nam	VN	VNM	704	VIET NAM	Dong	VND	704	0
Virgin Islands (British)	VG	VGB	092	VIRGIN ISLANDS (BRITISH)	US Dollar	USD	840	2
Virgin Islands (U.S.)	VI	VIR	850	VIRGIN ISLANDS (U.S.)	US Dollar	USD	840	2
Wallis and Futuna	WF	WLF	876	WALLIS AND FUTUNA	CFP Franc	XPF	953	0
Western Sahara*	EH	ESH	732	WESTERN SAHARA	Moroccan Dirham	MAD	504	2
Yemen	YE	YEM	887	YEMEN	Yemeni Rial	YER	886	2
Zambia	ZM	ZMB	894	ZAMBIA	Zambian Kwacha	ZMW	967	2
Zimbabwe	ZW	ZWE	716	ZIMBABWE	Zimbabwe Dollar	ZWL	932	2
				ZZ01_Bond Markets Unit European_EURCO*	Bond Markets Unit European Composite Unit (EURCO)	XBA	955	N.A.

ISO-3166 Country Codes				ISO 4217 Currency Codes				
				ZZ02_Bond Markets Unit European_EMU-6*	Bond Markets Unit European Monetary Unit (E.M.U.-6)	XBB	956	N.A.
				ZZ03_Bond Markets Unit European_EUA-9*	Bond Markets Unit European Unit of Account 9 (E.U.A.-9)	XBC	957	N.A.
				ZZ04_Bond Markets Unit European_EUA-17*	Bond Markets Unit European Unit of Account 17 (E.U.A.-17)	XBD	958	N.A.
				ZZ06_Testing_Code*	Codes specifically reserved for testing purposes	XTS	963	N.A.
				ZZ07_No_Currency*	The codes assigned for transactions where no currency is involved	XXX	999	N.A.
				ZZ08_Gold*	Gold	XAU	959	N.A.
				ZZ09_Palladium*	Palladium	XPD	964	N.A.
				ZZ10_Platinum*	Platinum	XPT	962	N.A.

ISO-3166 Country Codes				ISO 4217 Currency Codes				
				ZZ11_Silver*	Silver	XAG	961	N.A.

1. Kosovo is not listed as an ISO standard country. The unofficial 3-digit codes, as used by Mastercard and others, is used in ActiveAccess until Kosovo is assigned an ISO code.

Sample Request Response

```
[com.gpayments.caas.client.CaasClient]\|p\|[16] verify registration request:
[card=[id=,

number=5564-26XX-XXXX-3312,

type=SPA,

cardName=<null>,

Context_Blob=],

transaction=[xid=<null>,

purchaseAmount=<null>,

purchaseCurrency=<null>,

purchaseDate=<null>,

purchaseDesc=<null>,

merchantId=123456789012345,

merchantName=<null>,

merchantUrl=<null>,

merchantCountry=<null>,

acqBin=412345,

cardHolderIp=<null>,

cardExpiry=<null>,

cvd=<null>,

issuerName=Any Bank,

threeDSProtocolVersion=1.0.2],

iv=<null>]\|-

[com.gpayments.caas.client.CaasClient]\|p\|[16] verify registration response:
[cardInfo={[cardID=556426013102312,
```

```
Context_Blob=595,

cardName=SPA card,

pam=<null>,

sis=<null>,

priSec=2,

regStatus=1,

authRequired=1,

authType=<null>,

regToken=<null>,

authTypeSup={1,2,3,4,5,9,10},

proofAttempt=<null>,

activationDuringShopping=true,

identityData=[data={[value=<null>,

error=<null>,

name=cname,

authType=<null>,

format=[a-zA-Z ]+,

mask=false,

confirm=<null>],[value=<null>,

error=<null>,

name=pin,

authType=<null>,

format=\\w+,

mask=true,

confirm=<null>]}]]},
```

```
code=1,

errorMessage=warning-default warning for default response code 1,

errorDetail=warning-default warning for default response code 1]\|-

[com.gpayments.caas.client.CaasClient]\|p\|[17] verify-identity request:
[purpose=2,

identityData=[data={[value=Joe Citizen,

error=<null>,

name=cname,

authType=<null>,

format=<null>,

mask=<null>,

confirm=<null>],[value=123456,

error=<null>,

name=pin,

authType=<null>,

format=<null>,

mask=<null>,

confirm=<null>]}],

card=[id=5564260131023312,

number=5564-26XX-XXXX-3312,

type=SPA,

cardName=<null>,

Context_Blob=595],

transaction=[xid=MDAwMDAwMDAwMDAwMDAwMDAxMDA=,

purchaseAmount=12365,

purchaseCurrency=840,
```

```
purchaseDate=[eon=<null>,

year=2016,

month=11,

day=1,

timezone=210,

hour=7,

minute=50,

second=12,

fractionalSecond=0.000],

purchaseDesc=Blue shirt,

merchantId=123456789012345,

merchantName=Test Merchant,

merchantUrl=https://www.testmerchant.com/,

merchantCountry=231,

acqBin=412345,

cardHolderIp=192.168.1.100,

cardExpiry=2412,

cvd=<null>,

issuerName=Any Bank,

threeDSProtocolVersion=1.0.2],

iv=<null>]\|-

[com.gpayments.caas.client.CaasClient]\|p\|[17] verify-identity response:
[code=0,

identityData=<null>,

authData=[data={[value=<null>,

error=<null>,
```

```
name=password,

authType=1,

format=<null>,

mask=true,

confirm=true],[value=<null>,

error=<null>,

name=mobileNo,

authType=2,

format=<null>,

mask=<null>,

confirm=true],[value=<null>,

error=<null>,

name=token,

authType=2,

format=<null>,

mask=<null>,

confirm=<null>],[value=<null>,

error=<null>,

name=token,

authType=3,

format=<null>,

mask=<null>,

confirm=<null>],[value=<null>,

error=<null>,

name=token,

authType=4,
```

```
format=<null>,

mask=<null>,

confirm=<null>],[value=<null>,

error=<null>,

name=token,

authType=5,

format=<null>,

mask=<null>,

confirm=<null>],[value=<null>,

error=<null>,

name=serialNo,

authType=3,

format=<null>,

mask=<null>,

confirm=true],[value=<null>,

error=<null>,

name=serialNo,

authType=4,

format=<null>,

mask=<null>,

confirm=true],[value=<null>,

error=<null>,

name=serialNo,

authType=5,

format=<null>,
```

```
mask=<null>,

confirm=true],[value=<null>,

error=<null>,

name=userId,

authType=9,

format=<null>,

mask=<null>,

confirm=<null>],[value=<null>,

error=<null>,

name=password,

authType=9,

format=<null>,

mask=true,

confirm=true],[value=81z53x,

error=<null>,

name=challenge,

authType=10,

format=\\w+,

mask=<null>,

confirm=<null>],[value=<null>,

error=<null>,

name=response,

authType=10,

format=<null>,

mask=<null>,

confirm=<null>]}],
```

```
errorMessage=success,

errorDetail=success]\|-

[com.gpayments.caas.client.CaasClient]\|p\|[20] register request:
[registerData=[data={[value=123456,

error=<null>,

name=password,

authType=1,

format=<null>,

mask=<null>,

confirm=<null>]}],

card=[id=5564260131023312,

number=5564-26XX-XXXX-3312,

type=SPA,

cardName=<null>,

Context_Blob=595],

transaction=[xid=MDAwMDAwMDAwMDAwMDAwMDAxMDA=,

purchaseAmount=12365,

purchaseCurrency=840,

purchaseDate=[eon=<null>,

year=2016,

month=11,

day=1,

timezone=210,

hour=7,

minute=52,

second=55,
```

```
fractionalSecond=0.000],

purchaseDesc=Blue shirt,

merchantId=123456789012345,

merchantName=Test Merchant,

merchantUrl=https://www.testmerchant.com/,

merchantCountry=231,

acqBin=412345,

cardHolderIp=192.168.1.100,

cardExpiry=2412,

cvd=<null>,

issuerName=Any Bank,

threeDSProtocolVersion=1.0.2],

iv=<null>]\|-

[http-nio-8080-exec-1] [com.gpayments.caas.client.CaasClient]\|p\|[20] register
response: [code=0,

registerData=<null>,

errorMessage=success,

errorDetail=success]\|-
```

ActiveAccess\files\acs\acs.war\WEB-INF\lib\caas.client-<(version)>.jar

- caas.server.wsdl
- schema.xsd

SMS via JMS

This section is for issuer banks that have already implemented their own SMS sender module and require ActiveAccess to be integrated to support that module via JMS.

An Overview of SMPP

The Short Message Peer-to-Peer (SMPP) in telecommunications terms, refers to an open, industry standard protocol designed to provide a flexible data communication interface for the transfer of short message data between External Short Messaging Entities (ESME), Routing Entities (RE) and Message Centres.

SMPP Operation

The protocol is based on pairs of request/response PDUs (Protocol Data Units, or packets) exchanged over TCP connections. PDUs are binary encoded for efficiency. Data exchange may be synchronous, where each peer waits for a response for each PDU being sent, or asynchronous, where multiple requests can be issued without waiting and acknowledged in a skew order by the other peer. The number of unacknowledged requests is called a window; for the best performance both communicating sides must be configured with the same window size.

SMPP Versions

The SMPP standard has evolved during its time. The most commonly used versions of SMPP are:

- SMPP 3.3 the oldest used version; supports GSM only
- SMPP 3.4 adds Tag-Length-Value (TLV) parameters, support of non-GSM SMS technologies and the transceiver support (single connections that can send and receive messages)
- SMPP 5.0 is the latest version of SMPP; adds support for cell broadcasting

The applicable version is passed in the interface_version parameter of a bind command.

ActiveAccess supports SMPP-API-0.3.9.1 for communication with SMSC.

PDU Format

SMPP PDU starts with a header, followed by the body:

SMPP PDU				
PDU Header (mandatory)				PDU Body (Optional)
Command length	Command Id	Command Status	Sequence Id	PDU Body
4 octets		Length = (Command Length value - 4) octets		

PDU Header

Each PDU starts with a header. The header consists of 4 fields, each with a length of 4 octets:

command_length: Is the overall length of the PDU in octets (including command_length field itself); must be ≥ 16 as each PDU must contain the 16 octet header

command_id: Identifies the SMPP operation (or command)

command_status: Has always value of 0 in requests; in responses it carries information about the result of the operation

sequence_number: Is used to correlate requests and responses within an SMPP session; allows asynchronous communication (using a sliding window method)

All numeric fields in SMPP use the big endian order, which means that the first octet is the Most Significant Byte (MSB).

Example

This is an example of the binary encoding of a 60-octet submit_sm PDU. The data is shown in Hex octet values as a single dump and followed by a header and body break-down of that PDU.

This is best compared with the definition of the submit_sm PDU from the SMPP specification, in order to understand how the encoding matches the field by field definition.

The value break-down is shown with decimals in parentheses followed by the corresponding Hex values. When you see one or several hex octets appended, this represents the given field size that uses one or more octets encoding.

Again, reading the definition of the submit_sm PDU from the specification will make this clearer.

PDU Header

```
'command_length', (60) ... 00 00 00 3C

'command_id', (4) ... 00 00 00 04

'command_status', (0) ... 00 00 00 00

'sequence_number', (5) ... 00 00 00 05
```

PDU Body

```
'service_type', () ... 00

'source_addr_ton', (2) ... 02

'source_addr\_[npi](http://en.wikipedia.org/wiki/Numbering_plan)', (8) ... 08

'source_addr', (555) ... 35 35 35 00

'dest_addr_ton', (1) ... 01

'dest_addr\_[npi](http://en.wikipedia.org/wiki/Numbering_plan)', (1) ... 01

'dest_addr', (555555555) ... 35 35 35 35 35 35 35 35 35 00

'esm_class', (0) ... 00

'protocol_id', (0) ... 00

'priority_flag', (0) ... 00

'schedule_delivery_time', (0) ... 00

'validity_period', (0) ... 00

'registered_delivery', (0) ... 00

'replace_if_present_flag', (0) ... 00

'data_coding', (0) ... 00
```

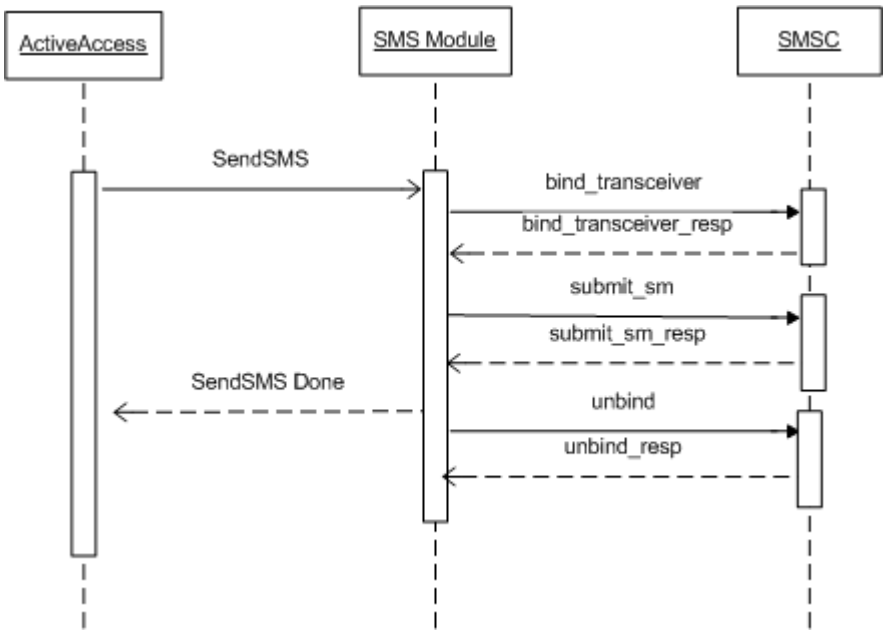
```
'sm_default_msg_id', (0) ... 00

'sm_length', (15) ... 0F

'short_message', (Hello wikipedia) ... 48 65 6C 6C 6F 20 77 69 6B 69 70 65 64 69
61'
```

An overview of the ActiveAccess SMS module

ActiveAccess has implemented an SMS module that provides the ability to send OTP SMS's to cardholders during the authentication process. This module uses SMPP v3.9 for its communication to SMSC. The following diagram illustrates a typical SMPP request/response sequence between an SMSC and ActiveAccess, bound as a Transceiver.



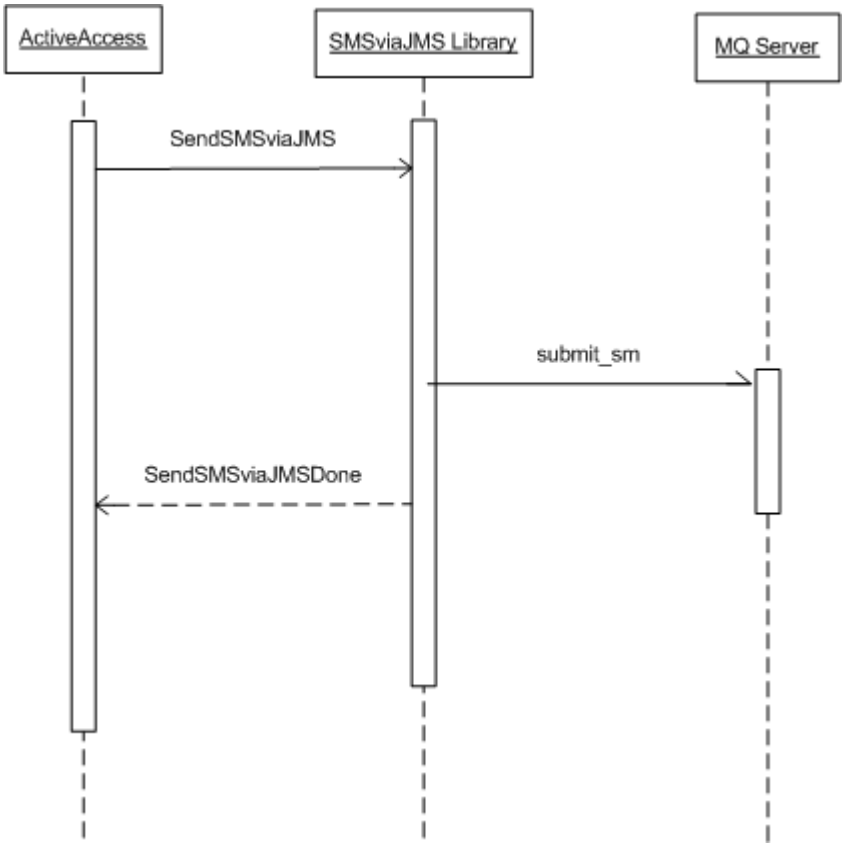
SMS via JMS (MQ Server) Solution

SMPP is an interactive messaging process, which is based on a pair of requests/responses. As MQ only accepts messages as a datagram, no response will be sent back to the sender. To integrate ActiveAccess with MQ Server for sending OTP SMS, GPayments has developed an **SMS via JMS library**.

The **SMS via JMS library** acts as a real SMPP client but it only sends **submit_sm** requests to MQ Server without any bind/unbind/enqlink commands before and after the submit_sm.

As MQ server will not be a real SMSC, as well, it only needs to implement a service that picks up the submit_sm PDUs and parses the required/optional parameters and then consumes them.

The following diagram illustrates an SMS via JMS request/response sequence between ActiveAccess, **SMS via JMS library** and MQ Server.



> sms-smpp-jms - lib.png

Format of submit_sm Message Request

MQ Server, as the server side of the SMS service, receives and queues submit_sm request. This request is a byte array. The following table provides specific details of a submit_sm request:

> **Note**
>
> ActiveAccess sets values for parameters highlighted in blue. So the MQ Service Provider only needs to refer to the highlighted parameters.

> **Note**
>
> Optional parameters which will be used for SMSs longer than 254 bytes have been highlighted in green.

Header

Field Name	Size (octets)	Type	Description
command_length	4	Integer	Set to overall length of PDU.
command_id	4	Integer	submit_sm (0x00000004)
command_status	4	Integer	Set to NULL
sequence_number	4	Integer	Set to a Unique sequence number. The associated submit_sm_resp PDU will echo this sequence number.

Mandatory Parameters

Field Name	Size (octets)	Type	Description
service_type	Var. Max 6	C-Octet String	The service_type parameter can be used to indicate the SMS Application service associated with the message. Specifying the service_type allows the ESME to: enhance messaging service, such as "replace by service" type control the teleservice used on the air interface. Set to NULL for default SMSC settings.
source_addr_ton	1	Integer	Type of Number for source address. It's set to Alphanumeric (5) by ActiveAccess
source_addr_npi	1	Integer	Numbering Plan Indicator for source address. It's set to Unknown (0) by ActiveAccess
source_addr	Var. Max 21	C-Octet String	Address of the sender that originates this message. If this is not known, it is set to NULL (Unknown). If there is no input, it will be set to default (NULL)
dest_addr_ton	1	Integer	Number Type for destination address. It's set to Unknown (0) by ActiveAccess

Field Name	Size (octets)	Type	Description
dest_addr_npi	1	Integer	Numbering Plan Indicator for destination. It's set to Unknown (0) by ActiveAccess
destination_addr	Var. Max 21	C-Octet String	Destination address of this short message. For mobile terminated messages, this is the directory number of the recipient MS.
esm_class	1	Integer	Indicates the Message Mode & Message Type.
protocol_id	1	Integer	Protocol Identifier. Network specific field.
priority_flag	1	Integer	Designates the priority level of the message.
schedule_delivery_time	1 or 17	C-Octet String	The short message is to be scheduled by the SMSC for delivery. Set to NULL for immediate message delivery.
validity_period	1 or 17	C-Octet String	The validity period of this message. Set to NULL to request the SMSC default validity period.
registered_delivery	1	Integer	Indicator to signify if an SMSC delivery receipt or an SME acknowledgement is required.
replace_if_present_flag	1	Integer	Flag, to indicate if submitted message should replace an existing message.
data_coding	1	Integer	Defines the encoding scheme of the short message user data. If message contains alphanumeric characters and no Unicode characters, data_coding is set to Default GSM Alphabet with extension (0) If message contains Unicode characters, data_coding is set to UCS2 (8)
sm_default_msg_id	1	Integer	Indicates the short message to be sent from a list of pre-defined ('canned') short messages stored on the SMSC. If a SMSC canned message is not in use, default value is set to NULL.
sm_length	1	Integer	Length in octets of the short_message user data.

Field Name	Size (octets)	Type	Description
short_message	Var. 0-254	Octet String	Up to 254 octets of short message user data. The exact physical limit for short_message size may vary according to the underlying network. Applications which need to send messages longer than 254 octets should use the message_payload parameter. In this case the sm_length field should be set to zero. Note: The short message data should be inserted in either the short_message or message_payload fields. Both fields must not be used simultaneously.

Optional Parameters

Field Name	Tag	Size (octets)	Type	Description
user_message_reference	0x0204	2	Integer	ESME assigned message reference number.
source_port	0x020A	2	Integer	Indicates the application port number associated with the source address of the message. This parameter should be present for WAP applications.
source_addr_subunit	0x000D	1	Integer	Indicates the subcomponent in the destination device which created the user data.
destination_port	0x020B	2	Integer	Indicates the application port number associated with the destination address of the message. This parameter should be present for WAP applications.
dest_addr_subunit	0x0005	1	Integer	The subcomponent in the destination device for which the user data is intended.
sar_msg_ref_num	0x020C	2	Integer	The reference number for a particular concatenated short message.

Field Name	Tag	Size (octets)	Type	Description
sar_total_segments	0x020E	1	Integer	Indicates the total number of short messages within the concatenated short message.
sar_segment_seqnum	0x020F	1	Integer	Indicates the sequence number of a particular short message fragment within the concatenated short message.
more_messages_to_send	0x0426	1	Integer	Indicates that there are more messages to follow for the destination SME.
payload_type	0x0019	1	Integer	Defines the type of payload (e.g. WDP, WCMP, etc.).
message_payload	 0x0424	variable	Octet String	Contains the extended short message user data. Up to 64K octets can be transmitted. Note: The short message data should be inserted in either the short_message or message_payload fields. Both fields should not be used simultaneously. The sm_lengthfield should be set to zero if using the message_payload parameter
privacy_indicator	0x0201	1	Integer	Indicates the level of privacy associated with the message.
callback_num	0x0381	var 4-19	Octet String	A callback number associated with the short message. This parameter can be included as a number of the times for multiple callback addresses.
callback_num_pres_ind	0x0302	1	Bit mask	Defines the callback number presentation and screening. If this parameter is present and there are multiple instances of the callback_num parameter, then this parameter must occur an equal number of instances and the order of occurrence determines the particular callback_num_pres_ind which corresponds to a particular callback_num.

Field Name	Tag	Size (octets)	Type	Description
callback_num_atag	0x0303	var max 65	Octet String	Associated to a displayable alphanumeric tag with the callback number. If this parameter is present and there are multiple instances of the callback_num parameter then this parameter must occur an equal number of instances and the order of occurrence determines the particular callback_num_atag which corresponds to a particular callback_num.
source_subaddress	0x0202	var 2-23	Octet String	The subaddress of the message originator.
dest_subaddress	0x0203	var 2-23	Octet String	The subaddress of the message destination.
user_response_code	0x0205	1	Integer	A user response code. The actual response codes are implementation specific.
display_time	0x1201	1	Integer	Provides the receiving MS with a display time associated with the message
sms_signal	0x1203	2	Integer	Indicates the alerting mechanism when the message is received by an MS.
ms_validity	0x1204	1	Integer	Indicates validity information for this message to the recipient MS.
ms_msg_wait_facilities	0x0030	1	Bit mask	This parameter controls the indication and specifies the message type (of the message associated with the MWI) at the mobile station.
number_of_messages	0x0304	1	Integer	Indicates the number of messages stored in a mail box
alert_on_msg_delivery	0x130C	0	Integer	Request for an MS alert signal to be invoked on message delivery. No value part associated to this parameter.

Field Name	Tag	Size (octets)	Type	Description
language_indicator	0x020D	1	Integer	Indicates the language of an alphanumeric text message.
its_reply_type	0x1380	1	Integer	The MS user's reply method to an SMS delivery message received from the network is indicated and controlled by this parameter.
Its_session_info	0x1383	2	Octet String	Session control information for Interactive Teleservice
ussd_service_op	0x0501	1	Octet String	This parameter is used to identify the required USSD Service type when interfacing to a USSD system.
dest_network_type	0x0006	1	Integer	The correct network for the destination device.
dest_telematics_id	0x0008	2	Integer	The telematics identifier associated with the destination.
source_network_type	0x000E	1	Integer	The correct network associated with the originating device.
source_bearer_type	0x000F	1	Integer	The correct bearer type for delivering the user data to the destination.
source_telematics_id	0x0010	1	Integer	The telematics identifier associated with the source.
qos_time_to_live	0x0017	4	Integer	Time to live as a relative time in seconds from submission.
additional_status_info_text	0x001D	var 1-256	C Octet String	ASCII text giving a description of the meaning of the response.
receipted_message_id	0x001E	var 1-65	C Octet String	SMSC message ID of receipted message. Should be present for SMSC Delivery Receipts and Intermediate Notifications.

Field Name	Tag	Size (octets)	Type	Description
SC_interface_version	0x0210	1	Integer	SMPP version supported by SMSC.
dpf_result	0x0420	1	Integer	Indicates whether the Delivery Pending Flag was set.
set_result	0x0421	1	Integer	Indicator for setting Delivery Pending Flag on delivery failure.
ms_availability_status	0x0422	1	Integer	The status of the mobile station.
network_error_code	0x0423	3	Octet String	Network Error Code. May be present for Intermediate Notifications and SMSC Delivery Receipts.
delivery_failure_reason	0x0425	1	Integer	Network Error Code. May be present for Intermediate Notifications and SMSC Delivery Receipts.
message_state	0x0427	1	Integer	Network Error Code. May be present for Intermediate Notifications and SMSC Delivery Receipts.
clientId	0x1400	15	Octet String	Network Error Code. May be present for Intermediate Notifications and SMSC Delivery Receipts.

Configuration

The **SMS via JMS library** reads the required parameters from a config file (sms_jms_config.properties) at start up. This config file will be loaded from AA_HOME directory where ActiveAccess configuration files have been installed.

It is possible to configure as many different SMSviaJMS clients as required for ActiveAccess. For each client, admin needs to add a unique prefix as JMS Client name to its configuration parameters in sms_jms_config.properties file: For Example, JMSClient1.

Then it is required to add a new SMS Centre in **MIA > System Management > Device Management > Edit Default Device Parameter-SMS > SMS Centre > New SMS Centre** with the following parameters:

Name: "JMSClient1"

Domain/IP: "SMSviaJMS"

"SMSviaJMS" determines for ActiceAccess to use JMS channel and load JMSClient1 properties for connecting to MQ Server.

JMSClient1.MQ_SERVER_SECURE_CHANNEL

This parameter specifies the type of channel (PLAIN or SSL) between **SMS via JMS library** and MQ Server.

Default: FALSE (PLAIN)

Example: JMSClient1.MQ_SERVER_SECURE_CHANNEL=TRUE

JMSClient1.MQ_SERVER_CLIENT_AUTH

If JMSClient1.MQ_SERVER_SECURE_CHANNEL=TRUE, this parameter specifies that the MQ Server requires client authentication on secure channel or not.

Default: FALSE

Example: JMSClient1.MQ_SERVER_CLIENT_AUTH=TRUE

JMSClient1.MQ_CLIENT_TRUSTSTORE

If JMSClient1.MQ_SERVER_SECURE_CHANNEL=TRUE, this parameter specifies the path of the truststore with trusted root certificates of the queue manager.

Example: JMSClient1.MQ_SERVER_TRUSTSTORE=conf/trustcertificates.jks

JMSClient1.MQ_CLIENT_TRUSTSTORE_PASSWORD

If JMSClient1.MQ_SERVER_SECURE_CHANNEL=TRUE, this parameter specifies the password of the truststore.

Example: JMSClient1.MQ_SERVER_TRUSTSTORE_PASSWORD=123456

JMSClient1.MQ_CLIENT_KEYSTORE

If JMSClient1.MQ_SERVER_SECURE_CHANNEL=TRUE and JMSClient1.MQ_SERVER_CLIENT_AUTH=TRUE, this parameter specifies the path of the keystore that must contain required client certificate which is trusted by the queue manager.

Example: JMSClient1.MQ_KEYSTORE=

JMSClient1.MQ_CLIENT_KEYSTORE_PASSWORD

If JMSClient1.MQ_SERVER_SECURE_CHANNEL=TRUE, this parameter specifies the password of the client keystore.

Example: JMSClient1.MQ_KEYSTORE_PASSWORD=123456

JMSClient1.MQ_CLIENT_KEYTYPE

If JMSClient1.MQ_SERVER_SECURE_CHANNEL=TRUE, this parameter specifies one of the following as the client's keystore type:

- JKS
- JCEKS
- P12
- PKCS12

Example: JMSClient1.MQ_KEYSTORE_TYPE=JKS

JMSClient1.MQ_SSL_CIPHER_SUITE

If JMSClient1.MQ_SERVER_SECURE_CHANNEL=TRUE, this parameter restricts the type of cipher algorithm which is accepted for securing the SSL channel. Any other algorithm except this one which is specified by this parameter is rejected during the handshake. It is strongly recommended that TLSv1.1 or TLSv1.2 is used because CipherSpecs and CipherSuites, such as SSLv3, have known security vulnerabilities.

> **Note**
>
> For the complete list of supported CipherSpecs and CipherSuites, please see http://www.ibm.com/support/knowledgecenter/en/SSFKSJ_7.5.0/com.ibm.mq.sec.doc/q014260_.htm.

Example: MQ_SSL_CIPHER_SUITE=TLS_RSA_WITH_AES_256_CBC_SHA256

> **Note**
>
> This parameter is case sensitive.

JMSClient1.MQ_CHANNEL_NAME

Specifies the name of the channel, in which it will be used in order to connect to Queue Manager.

Example: JMSClient1.MQ_CHANNEL_NAME=CHANNEL1

> **Note**
>
> This parameter is case sensitive.

JMSClient1.MQ_QUEUE_MGR_NAME

Specifies the name of the Queue Manager.

Example: JMSClient1.MQ_QUEUE_MGR_NAME=QM1

> **Note**
>
> This parameter is case sensitive.

JMSClient1.MQ_QUEUE_NAME

Specifies the name of the Queue to put messages on.

Example: JMSClient1.QUEUE_NAME=LQ1

> **Note**
>
> This parameter is case sensitive.

JMSClient1.MQ_USE_USERID

Specifies whether JMSClient1.MQ_USERID is used for channel authentication or not.

Default: FALSE;

Example: JMSClient1.MQ_USE_USERID=TRUE

JMSClient1.MQ_USERID

If JMSClient1.MQ_USE_USERID=TRUE, the parameter specifies the JMSClient1.MQ_USERID which is required for channel authentication.

Example: JMSClient1.MQ_USERID=userId

> **Note**
>
> This parameter is case sensitive.

JMSClient1.MQ_USERID_PASSWORD

When MQ_USE_USERID=TRUE, this parameter specifies the password which is required for channel authentication.

Example: JMSClient1.MQ_USERID_PASSWORD= password

> **Note**
>
> This parameter is case sensitive.

JMSClient1.MQ_MAX_CONNECTION

Specifies the maximum allowed connections to MQ Server. If a limit is set and the library reaches that limit, incoming requests might encounter an MQException, with reason code of JMSClient1.MQRC_MAX_CONNS_LIMIT_REACHED

Default: 5

Example: JMSClient1.MQ_MAX_CONNECTION =10

JMSClient1.MQ_TIMEOUT

Specifies the amount of time that an idle connection is kept in pool after the specified time, the library ends the connection.

Default: 3600000

Example: JMSClient1.MQ_TIMEOUT=180000

JMSClient1.MQ_MAX_UNUSED_CONNECTION

Specifies the maximum number of idle connections

Default: 3

Example: JMSClient1.MQ_MAX_UNUSED_CONNECTION=6

JMSClient1.MQ_MSG_EXPIRY

Specifies the number of seconds that messages are kept in the queue before it expires. Default is set to **Never**, which ensures that the message is delivered and waits in the queue until it is retrieved, no matter how long it may take. To use the default value set it to zero.

Default: 0

Example: JMSClient1.MQ_MSG_EXPIRY=10

JMSClient1.MQ_MSG_PRIORITY

Specifies the priority of the message. By default, the message priority defaults to the default priority for the queue. If the queue uses priorities to order messages and this message should have a specific priority, specify the priority. Priorities can range from 1 (lowest) up to 9 (highest). To use the default priority of queue set the value to 0

Default: 0

Example: JMSClient1.MQ_MSG_PRIORITY=8

JMSClient1.MQ_MSG_DELIVERY

Specifies the delivery mode if the message (PERSISTENT or NOT PERSISTENT). Persistent messages are written to logs and queue data files. If a queue manager is restarted after a failure, it recovers these persistent messages as necessary from the logged data. Messages that are not persistent are discarded if a queue manager stops.

YES Use Persistent

NO Do not use persistent

DEF Use queue property as default

Default: DEF

Example: JMSClient1.MQ_MSG_DELIVERY=YES

JMSClient1.MQ_CONNECTION_NAME_LIST

This parameter is a comma-separated list that contains the host name and port information to be used to connect to a queue manager in client mode. SMS via JMS application attempts to connect to each host in list order. If the end of the connection name list is reached without establishing a successful connection, it throws the MQRC_QMGR_NOT_AVAILABLE with WebSphere MQ Reason Code in log file.

Example: JMSClient1.MQ_CONNECTION_NAME_LIST=127.0.0.1(1414), 192.168.1.133(1414), 192.168.1.100(1415)

JMSClient1.MQ_CCSID

This parameter specifies the code page for the SMS text. The list of codes is available at:

http://www-01.ibm.com/software/globalization/ccsid/ccsid_registered.html

If JMSClient1.MQ_CCSID is not set, code page of queue will be used as default value.

Some possible values are:

850 Commonly used ASCII codeset

819 The ISO standard ASCII codeset

37 The American EBCDIC codeset

1200 Unicode

1208 UTF-8

Example: JMSClient1.MQ_CCSID=1208

Submit_SM Processing

Following sections describe how Header and Body of the submit_sm message should be processed by MQ Service provider:

Header Processing

As mentioned in 1.1.3, PDU Header contains 4 parameters with octet length. To process the PDU message, the header should be decoded first:

- Get the Integer value of each part:

0 – 4 Specifies the cmd_len, which shows the length of received message.

4 – 8 Specifies the cmd_id, which will check to determine whether the received message is a submit_sm or not.

8 – 12 Specifies the cmd_status, which expects to be null for submit_sm

12 – 16 Specifies the seq_no, the sequence number of received message which will be used to find by ACS Session ID

Body processing

PDU Body:

16 – 22 -> **service_type**: which is null for default SMSC settings, so it is actually 16-17

22 – 23 -> **source_addr_ton:** If service-type has the max length it will be 22-23.

23 – 24 -> source_addr_npi

24 – 45 -> **source_addr:** Max length is 21, it will actually be 24 – source_addr length plus 1.

45 – 46 -> dest_addr_ton

46 – 47 -> dest_addr_npi

47 – 68 -> **destination_addr:** Max length is 21, it will actually be 47 – **destination_addr** length plus 1.

68 – 69 -> **esm_class**

69 - 70 -> **protocol_ID**

70- 71 -> priority_flag

71 – 88 -> schedule_delivery_time

88 – 105 -> validity_period

105 – 106 -> registered_delivery_flag

106 – 107 -> **data_coding**

107 – 108 -> sm_default_msg_id

108 - 109 -> **sm_length**

109 – 363 -> short_message

Optional parameter processing message_payload:

Field	Size (Octets)	Type	Description
Parameter Tag	2	Integer	message_payload
Length	2	Integer	Length of value part in octets
Value	variable	Octet String	Short message user data. The maximum size is SMSC and network implementation specific.

363 – 365 -> tag (message_payload = 1060 or 0x424)

365 – 367 -> len

367 – 64367 -> value

EXAMPLES

Example of submit_sm in form of byte array:

```
[0, 0, 0, -76, 0, 0, 0, 4, 0, 0, 0, 0, 0, 0, 0, 3, 0, 5, 0, 49, 48, 48, 50, 48,
0, 0, 0, 54, 49, 52, 50, 50, 52, 53, 50, 48, 53, 49, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, -125, 65, 99, 116, 105, 118, 97, 116, 105, 111, 110, 47, 82, 101, 103, 105,
115, 116, 114, 97, 116, 105, 111, 110, 32, 118, 105, 97, 32, 77, 73, 65, 58, 10,
49, 48, 32, 116, 111, 107, 101, 110, 10, 65, 58, 50, 54, 50, 55, 53, 57, 10, 66,
58, 56, 49, 55, 51, 57, 49, 10, 67, 58, 57, 54, 50, 49, 49, 56, 10, 68, 58, 49,
54, 54, 51, 56, 57, 10, 69, 58, 52, 49, 54, 50, 49, 54, 10, 70, 58, 50, 50, 51,
48, 52, 55, 10, 71, 58, 55, 55, 48, 57, 53, 52, 10, 72, 58, 54, 53, 52, 53, 50,
57, 10, 73, 58, 54, 52, 53, 55, 51, 56, 10, 74, 58, 54, 49, 56, 50, 50, 50]
```

> **Note**
>
> When retrieving String from byte array, it will be processed for **not null (≠ 0)** values

< PDU HEADER START >

0, 0, 0, -76, -> **cmd_len :** 180

0, 0, 0, 4, -> **cmd_id :** 4

0, 0, 0, 0, -> **cmd_status :** 0

0, 0, 0, 3, -> **seq_no :** 3

< PDU HEADER END >

0,5, 0, 49, 48, 48, -> **service_type :** "" (The length is 0 - **because the first element is null** - so from 5 to 48 will be processed again for the next element)

5, -> source_addr_ton : 5

0, -> source_addr_npi : 0

49, 48, 48, 50, 48, 0, 0, 0, 54, 49, 52, 50, 50, 52, 53, 50, 48, 53, 49, 0, 0, -> **destination_addr :** 10020 (The length is 5 so from 0 to 0 will be processed again for the next element)

0, -> dest_addr_ton : 0

0, -> dest_addr_npi : 0

54, 49, 52, 50, 50, 52, 53, 50, 48, 53, 49, 0, 0, 0, 0, 0, 0, 0, 0, 0, -> **destination_addr :** 61422452051

0, -> **esm_class** : 0

0, -> protocol_ID : 0

0, -> priority_flag : 0

0, 0, 0, 0, 0, 0, -125, 65, 99, 116, 105, 118, 97, 116, 105, 111, 110 -> **schedule_delivery_time :** ""

0, 0, 0, 0, 0, -125, 65, 99, 116, 105, 118, 97, 116, 105, 111, 110, 47, -> **validity_period** : ""

0, -> registered_delivery_flag : 0

0, -> replace_if_present_flag : 0

0, -> data_coding : 0

0, -> sm_default_msg_id : 0

-125, -> **sm_length :** 131

65, 99, 116, 105, 118, 97, 116, 105, 111, 110, 47, 82, 101, 103, 105, 115, 116, 114, 97, 116, 105, 111, 110, 32, 118, 105, 97, 32, 77, 73, 65, 58, 10, 49, 48, 32, 116, 111, 107, 101, 110, 10, 65, 58, 50, 54, 50, 55, 53, 57, 10, 66, 58, 56, 49, 55, 51, 57, 49, 10, 67, 58, 57, 54, 50, 49, 49, 56, 10, 68, 58, 49, 54, 54, 51, 56, 57, 10, 69, 58, 52, 49, 54, 50, 49, 54, 10, 70, 58, 50, 50, 51, 48, 52, 55, 10, 71, 58, 55, 55, 48, 57, 53, 52, 10, 72, 58, 54, 53, 52, 53, 50, 57, 10, 73, 58, 54, 52, 53, 55, 51, 56, 10, 74, 58, 54, 49, 56, 50, 50, 50 **short_message** : Activation/Registration via MIA:

10 token

A:262759

B:817391

C:962118

D:166389

E:416216

F:223047

G:770954

H:654529

I:645738

J:618222

< PDU BODY END >

Example of submit_sm in form of HEX dump:

```
000000B4:00000004:00000000:00000003:

00050031:30303230:00000036:31343232:
```

```
34353230:35310000:00000000:00000000:
83416374:69766174:696F6E2F:52656769:
73747261:74696F6E:20766961:204D4941:
3A0A3130:20746F6B:656E0A41:3A323632:
3735390A:423A3831:37333931:0A433A39:
36323131:380A443A:31363633:38390A45:
3A343136:3231360A:463A3232:33303437:
0A473A37:37303935:340A483A:36353435:
32390A49:3A363435:3733380A:4A3A3631:
38323232:
```

< PDU HEADER >

000000B4: -> **cmd_len :** 173 (hex to int)

00000004: -> **cmd_id :** 4 (hex to int)

00000000: -> **cmd_status :** 0 (hex to int)

00000003: -> **seq_no :** 3 (hex to int)

< PDU HEADER END >

< PDU BODY >

- 00050031: ->

  00 => **service_type:** 0 (hex to String)

  05 => **source_addr_ton :** 5 (hex to int)

  00 => **source_addr_npi :** 0 (hex to int)

  31 => **source_addr starts** (hex to String)

  1

- 30303230: -> rest of **source_addr** (hex to String)

  0020

- 00000036: ->

  00 => **source_addr ends** (hex to String)

  00 => **dest_addr_ton :** 0 (hex to int)

  00 => **dest_addr_npi :** 0 (hex to int)

  36 => **destination_addr :** 6 (hex to String)

- 33353200: ->

  31343232 => rest of **destination_addr :** 1422 (hex to String)

- 34353230: ->

  34353230 => rest of **destination_addr :** 4520 (hex to String)

- 35310000: ->

  3531 => rest of **destination_addr :** 51 (hex to String)

  00 => end of **destination_addr** (hex to String)

- Complete destination_addr: 61422452051

  00 => **esm_class :** 0 (hex to int)

- 00000000: ->

  00 => **protocol_ID :** 0 (hex to int)

  00 => **priority_flag :** 0 (hex to int)

  00 => **schedule_delivery_time :** 0 (hex to String)

  00 => **validity_period :** 0 (hex to String)

- 00000000: ->

  00 => registered_delivery_flag : 0 (hex to int)

  00 => replace_if_present_flag : 0 (hex to int)

  00 => **data_coding :** 0 (hex to int)

  00 => **sm_default_msg_id :** 0 (hex to int)

- 83416374: ->

  83 => **sm_length** : 131 (hex to int)

  416374 => **short_message starts** (hex to String)

69766174:696F6E2F:52656769: (hex to String)

73747261:74696F6E:20766961:204D4941: (hex to String)

3A0A3130:20746F6B:656E0A41:3A323632: (hex to String)

3735390A:423A3831:37333931:0A433A39: (hex to String)

36323131:380A443A:31363633:38390A45: (hex to String)

3A343136:3231360A:463A3232:33303437: (hex to String)

0A473A37:37303935:340A483A:36353435: (hex to String)

32390A49:3A363435:3733380A:4A3A3631: (hex to String)

38323232: -> **short_message ends** (hex to String)

Short_message : Activation/Registration via MIA:

10 token

A:262759

B:817391

C:962118

D:166389

E:416216

F:223047

G:770954

H:654529

I:645738

J:618222

Example of submit_sm in form of byte array for UCS-2 or Unicode SMS:

```
0, 0, 0, 87, 0, 0, 0, 4, 0, 0, 0, 0, 0, 0, 0, 3, 0, 5, 0, 49, 50, 51, 52, 53,
54, 55, 0, 0, 0, 54, 49, 52, 52, 50, 48, 50, 49, 52, 53, 50, 49, 52, 0,0 ,0, 0,0
, 0, 0, 0, 8, 0, 34, 0, 65, 0, 68, 0, 83, 0, 58, 0, 32, 4, 63, 0, 32, 4, 54, 0,
10, 0, 65, 0, 58, 0, 54, 0, 57, 0, 52, 0, 48, 0, 54, 0, 48
```

< PDU HEADER>

0, 0, 0, 87 -> **cmd_len** : 104

0, 0, 0, 4 -> **cmd_id** : 4

0, 0, 0, 0 -> **cmd_status** : 0

0, 0, 0, 3 -> **seq_no** : 3

< PDU HEADER END>

0, 5, 0, 49, 50, 51, -> **service_type** : "" (The length is 0 - **because the first element is null** – the rest will be processed again) :

< byte to String>

5, -> **source_addr_ton** : 5 < byte to Integer>

0, -> **source_addr_npi** : 0 < byte to Integer>

49, 50, 51, 52, 53, 54, 55, 0, 0, 0, 54, 49, 52, 52, 50, 48, 50, 49, 52, 53, 50, -> **source_addr** : 1234567 (The length is 7 so the rest will be processed again for the next element) < byte to String>

0, -> **dest_addr_ton** : 0 < byte to Integer>

0, -> **dest_addr_npi** : 0 < byte to Integer>

54, 49, 52, 52, 50, 48, 50, 49, 52, 53, 50, 49, 52, 0, 0 , 0, 0, 0, 0, 0, 0 -> **destination_addr** : 6144202145214 < byte to String>

0, -> **esm_class** : 0 < byte to Integer>

0, -> **protocol_ID** : 0 < byte to Integer>

0, -> **priority_flag** : 0 < byte to Integer>

0, 0, 0, 0, 8, 0, 34, 0, 65, 0, 68, 0, 83, 0, 58, 0, 32, -> schedule **_delivery_time** : "" < byte to String>

0, 0, 0, 8, 0, 34, 0, 65, 0, 68, 0, 83, 0, 58, 0, 32, 4 -> **validity_period** : "" < byte to String>

0, -> **registered_delivery_flag** : 0 < byte to Integer>

0, -> **replace_if_present_flag** : 0 < byte to Integer>

8, -> **data_coding** : 8 < byte to Integer>

0, -> **sm_default_msg_id** : 0 < byte to Integer>

34, -> **sm_length** : 34 < byte to Integer>

0, 65, 0, 68, 0, 83, 0, 58, 0, 32, 4, 63, 0, 32, 4, 54, 0, 10, 0, 65, 0, 58, 0, 54, 0, 57, 0, 52, 0, 48, 0, 54, 0, 48 **short_message** : ADS: п ж

A:694060

< byte to String with fix length>

< PDU BODY END>

Example of submit_sm in form of HEX dump for UCS-2 or Unicode SMS:

```
00000057:00000004:00000000:00000003:
00050031:32333435:36370000:00363134:
34323032:31343532:31340000:00000000:
00000800:22004100:44005300:3A002004:
3F002004:36000A00:41003A00:36003900:
34003000:360030
```

< PDU HEADER >

00000057: -> **cmd_len** : 87 (hex to int)

00000004: -> **cmd_id** : 4 (hex to int)

00000000: -> **cmd_status** : 0 (hex to int)

00000003: -> **seq_no** : 3 (hex to int)

< PDU HEADER END >

- 00050031: ->

  00 => **service_type**:0 (hex to String)

  05 => **source_addr_ton** : 5 (hex to int)

00 => **source_addr_npi** : 0 (hex to int)

31 => **source_addr starts** : 1 (hex to String)

- 32333435: -> rest of **source_addr** : 2345 (hex to String)
- 36370000: ->

363700 =>rest of **source_addr and it ends** : 67 (hex to String)

00 => **dest_addr_ton** : 0 (hex to int)

- 00363134: ->

00 => **dest_addr_npi** : 0 (hex to int)

363134 => **destination_addr** : 614 (hex to String)

- 34323032: ->

34323032=> rest of **destination_addr** : 4202 (hex to String)

- 31343532: ->

31343532=> rest of **destination_addr** : 1452 (hex to String)

- 31340000 ->

313400 => rest of **destination_addr** and its **end** (hex to String)

Complete destination_addr: 6144202145214

00 => **esm_class** : 0 (hex to int)

- 00000000: ->

00 => **protocol_ID** : 0 (hex to int)

00 => **priority_flag** : 0 (hex to int)

00 => **schedule_delivery_time** : 0 (hex to String)

00 => **validity_period** : 0 hex to String)

- 00000800: ->

00 => **registered_delivery_flag** : 0 (hex to int)

00 => **replace_if_present_flag** : 0 (hex to int)

08 => **data_coding** : 8 (hex to int)

00 => **sm_default_msg_id** : 0 (hex to int)

- 22004100: ->

22 => **sm_length** : 34 (hex to int)

004100=> **short_message starts** (hex to String)

44005300:3A002004:

3F002004:36000A00:41003A00:36003900:

34003000:360030 **short_message ends** (hex to String)

Short_message : ADS: п ж

A:694060

< PDU BODY END>

Example of submit_sm in form of byte array for long SMS:

```
0, 0, 1, 127, 0, 0, 0, 4, 0, 0, 0, 0, 109, -63, 106, -6, 0, 5, 0, 77, 81, 83,
109, 115, 83, 101, 110, 100, 101, 114, 0, 0, 0, 54, 49, 52, 52, 54, 53, 50, 49,
52, 53, 50, 49, 52, 53, 50, 0, 0, 0, 0, 0, 0, 0, 0, 8, 0, 0, 4, 36, 1, 64, 0,
65, 0, 99, 0, 116, 0, 105, 0, 118, 0, 97, 0, 116, 0, 105, 0, 111, 0, 110, 0, 47,
0, 82, 0, 101, 0, 103, 0, 105, 0, 115, 0, 116, 0, 114, 0, 97, 0, 116, 0, 105, 0,
111, 0, 110, 0, 32, 0, 118, 0, 105, 0, 97, 0, 32, 0, 77, 0, 73, 0, 65, 0, 58, 0,
13, 0, 10, 0, 49, 0, 48, 0, 32, 0, 84, 0, 111, 0, 107, 0, 101, 0, 110, 0, 115,
0, 32, 0, 115, 0, 101, 0, 110, 0, 116, 0, 13, 0, 10, 0, 65, 0, 58, 0, 56, 0, 49,
0, 56, 0, 48, 0, 57, 0, 56, 0, 13, 0, 10, 0, 66, 0, 58, 0, 36, 0, 84, 0, 111, 0,
107, 0, 101, 0, 110, 0, 66, 0, 13, 0, 10, 0, 67, 0, 58, 0, 36, 0, 84, 0, 111, 0,
107, 0, 101, 0, 110, 0, 67, 0, 13, 0, 10, 0, 68, 0, 58, 0, 36, 0, 84, 0, 111, 0,
107, 0, 101, 0, 110, 0, 68, 0, 13, 0, 10, 0, 69, 0, 58, 0, 36, 0, 84, 0, 111, 0,
107, 0, 101, 0, 110, 0, 69, 0, 13, 0, 10, 0, 70, 0, 58, 0, 36, 0, 84, 0, 111, 0,
107, 0, 101, 0, 110, 0, 70, 0, 13, 0, 10, 0, 71, 0, 58, 0, 36, 0, 84, 0, 111, 0,
107, 0, 101, 0, 110, 0, 71, 0, 13, 0, 10, 0, 72, 0, 58, 0, 36, 0, 84, 0, 111, 0,
107, 0, 101, 0, 110, 0, 72, 0, 13, 0, 10, 0, 73, 0, 58, 0, 36, 0, 84, 0, 111, 0,
107, 0, 101, 0, 110, 0, 73, 0, 13, 0, 10, 0, 74, 0, 58, 0, 36, 0, 84, 0, 111, 0,
107, 0, 101, 0, 110, 0, 74, 0, 13, 0, 10, 6, 127
```

< PDU HEADER>

0, 0, 1, 127 -> cmd_len : 383

0, 0, 0, 4 -> cmd_id : 4

0, 0, 0, 0 -> cmd_status : 0 < PDU BODY END>

109, -63, 106, -6 -> seq_no : 1841392378

< PDU HEADER END>

0, 5, 0, 77, 81, 83, -> **service_type** : "" (The length is 0 - **because the first element is null** – the rest will be processed again) :

< byte to String>

5, -> **source_addr_ton** : 5 < byte to Integer>

0, -> **source_addr_npi** : 0 < byte to Integer>

77, 81, 83, 109, 115, 83, 101, 110, 100, 101, 114, 0, 0, 0, 54, 49, 52, 52, 54, 53, 50, -> **source_addr** : MQSMSSender (The length is 7 so the rest will be processed again for the next element) < byte to String>

0, -> **dest_addr_ton** : 0 < byte to Integer>

0, -> **dest_addr_npi** : 0 < byte to Integer>

54, 49, 52, 52, 54, 53, 50, 49, 52, 53, 50, 49, 52, 53, 50, 0, 0, 0, 0, 0, 0, -> **destination_addr** : 614465214521452 < byte to String>

0, -> **esm_class** : 0 < byte to Integer>

0, -> **protocol_ID** : 0 < byte to Integer>

0, -> **priority_flag** : 0 < byte to Integer>

0, 0, 0, 0, 8, 0, 0, 4, 36, 1, 64, 0, 65, 0, 99, 0, 116, -> **schedule_delivery_time** : "" v < byte to String>

0, 0, 0 , 8, 0, 0, 4, 36, 1, 64, 0, 65, 0, 99, 0, 116, 0, -> **validity_period** : "" < byte to String>

0, **registered_delivery_flag** : 0

0, **replace_if_present_flag** : 0

8, **data_coding** : 8

0, **sm_default_msg_id** : 0

0, **sm_length** : 0

short_message : "" because sm_length is 0

4, 36 **tag** : 1060

1, 64, **len**: 320

0, 65, 0, 99, 0, 116, 0, 105, 0, 118, 0, 97, 0, 116, 0, 105, 0, 111, 0, 110, 0, 47, 0, 82, 0, 101, 0, 103, 0, 105, 0, 115, 0, 116, 0, 114, 0, 97, 0, 116, 0, 105, 0, 111, 0, 110, 0, 32, 0, 118, 0, 105, 0, 97, 0, 32, 0, 77, 0, 73, 0, 65, 0, 58, 0, 13, 0, 10, 0, 49, 0, 48, 0, 32, 0, 84, 0, 111, 0, 107, 0, 101, 0, 110, 0, 115, 0, 32, 0, 115, 0, 101, 0, 110, 0, 116, 0, 13, 0, 10, 0, 65, 0, 58, 0, 56, 0, 49, 0, 56, 0, 48, 0, 57, 0, 56, 0, 13, 0, 10, 0, 66, 0, 58, 0, 36, 0, 84, 0, 111, 0, 107, 0, 101, 0, 110, 0, 66, 0, 13, 0, 10, 0, 67, 0, 58, 0, 36, 0, 84, 0, 111, 0, 107, 0, 101, 0, 110, 0, 67, 0, 13, 0, 10, 0, 68, 0, 58, 0, 36, 0, 84, 0, 111, 0, 107, 0, 101, 0, 110, 0, 68, 0, 13, 0, 10, 0, 69, 0, 58, 0, 36, 0, 84, 0, 111, 0, 107, 0, 101, 0, 110, 0, 69, 0, 13, 0, 10, 0, 70, 0, 58, 0, 36, 0, 84, 0, 111, 0, 107, 0, 101, 0, 110, 0, 70, 0, 13, 0, 10, 0, 71, 0, 58, 0, 36, 0, 84, 0, 111, 0, 107, 0, 101, 0, 110, 0, 71, 0, 13, 0, 10, 0, 72, 0, 58, 0, 36, 0, 84, 0, 111, 0, 107, 0, 101, 0, 110, 0, 72, 0, 13, 0, 10, 0, 73, 0, 58, 0, 36, 0, 84, 0, 111, 0, 107, 0, 101, 0, 110, 0, 73, 0, 13, 0, 10, 0, 74, 0, 58, 0, 36, 0, 84, 0, 111, 0, 107, 0, 101, 0, 110, 0, 74, 0, 13, 0, 10, 6, 127 value:

Activation/Registration via MIA:

10 Tokens sent

A:818098

B:$TokenB

C:$TokenC

D:$TokenD

E:$TokenE

F:$TokenF

G:$TokenG

H:$TokenH

I:$TokenI

J:$TokenJ

Example of submit_sm in form of HEX dump for long SMS:

```
0000017F:00000004:00000000:6DC16AFA:

0005004D:51536D73:53656E64:65720000:

00363134:34363532:31343532:31343532:
```

```
00000000:00000000:08000004:24014000:
41006300:74006900:76006100:74006900:
6F006E00:2F005200:65006700:69007300:
74007200:61007400:69006F00:6E002000:
76006900:61002000:4D004900:41003A00:
0D000A00:31003000:20005400:6F006B00:
65006E00:73002000:73006500:6E007400:
0D000A00:41003A00:38003100:38003000:
39003800:0D000A00:42003A00:24005400:
6F006B00:65006E00:42000D00:0A004300:
3A002400:54006F00:6B006500:6E004300:
0D000A00:44003A00:24005400:6F006B00:
65006E00:44000D00:0A004500:3A002400:
54006F00:6B006500:6E004500:0D000A00:
46003A00:24005400:6F006B00:65006E00:
46000D00:0A004700:3A002400:54006F00:
6B006500:6E004700:0D000A00:48003A00:
24005400:6F006B00:65006E00:48000D00:
0A004900:3A002400:54006F00:6B006500:
6E004900:0D000A00:4A003A00:24005400:
6F006B00:65006E00:4A000D00:0A067F
```

< PDU HEADER >

0000017F: -> **cmd_len** : 383 (hex to int)

00000004: -> **cmd_id** : 4 (hex to int)

00000000: -> **cmd_status** : 0 (hex to int)

6DC16AFA: -> **seq_no** : 1841392378 (hex to int)

< PDU HEADER END >

- 0005004D: ->

  00 => **service_type**:0 (hex to String)

  05 => **source_addr_ton** : 5 (hex to int)

  00 => **source_addr_npi** : 0 (hex to int)

  4D => **source_addr starts** : 1 (hex to String)
- 51536D73: -> rest of **source_addr** : QSms (hex to String)
- 53656E64: -> rest of **source_addr** : Send (hex to String)
- 65720000: ->

  6572 00=> rest of **source_addr** : er (hex to String)

  00 => **dest_addr_ton** : 0 (hex to int)
- 00363134: ->

  00 => **dest_addr_npi** : 0 (hex to int)

  363134 => **destination_addr** : 614 (hex to String)
- 34363532 -> rest of **destination_addr** :4652 (hex to String)
- 31343532: -> rest of **destination_addr** : 1452 (hex to String)
- 31343532 -> rest of **destination_addr** : 1452 (hex to String)

  313400 => rest of **destination_addr** and its **end** (hex to String)

  Complete **destination_addr** : 614465214521452
- 00000000: ->

  00 => end of **destination_addr** (hex to String)

  00 => **esm_class** : 0 (hex to int)

  00 => **protocol_ID** : 0 (hex to int)

  00 => **priority_flag** : 0 (hex to int)
- 00000000: ->

00 => **schedule_delivery_time** : 0 (hex to String)

00 => **validity_period** : 0 (hex to String)

00 => **registered_delivery_flag** : 0 (hex to int)

00 => **replace_if_present_flag** : 0 (hex to int)

- 08000004: ->

08 => **data_coding** : 8 (hex to int)

00 => **sm_default_msg_id** : 0 (hex to int)

00 => **sm_length** : 0 (hex to int)

=> **short_message** : ""because sm_length is 0, this parameter will not be processed

04 => process for message_payload tag

- 24014000: ->

24 => rest of message_payload tag : 1060

0140 => message_payload len : 320

00 => message_payload value **starts**

41006300:74006900:76006100:74006900:

6F006E00:2F005200:65006700:69007300:

74007200:61007400:69006F00:6E002000:

76006900:61002000:4D004900:41003A00:

0D000A00:31003000:20005400:6F006B00:

65006E00:73002000:73006500:6E007400:

0D000A00:41003A00:38003100:38003000:

39003800:0D000A00:42003A00:24005400:

6F006B00:65006E00:42000D00:0A004300:

3A002400:54006F00:6B006500:6E004300:

0D000A00:44003A00:24005400:6F006B00:

65006E00:44000D00:0A004500:3A002400:

54006F00:6B006500:6E004500:0D000A00:

46003A00:24005400:6F006B00:65006E00:

46000D00:0A004700:3A002400:54006F00:

6B006500:6E004700:0D000A00:48003A00:

24005400:6F006B00:65006E00:48000D00:

0A004900:3A002400:54006F00:6B006500:

6E004900:0D000A00:4A003A00:24005400:

6F006B00:65006E00:4A000D00:0A067F -> **message_payload** value **ends** (hex to String)

Activation/Registration via MIA:

10 Tokens sent

A:818098

B:$TokenB

C:$TokenC

D:$TokenD

E:$TokenE

F:$TokenF

G:$TokenG

H:$TokenH

I:$TokenI

J:$TokenJ

< PDU BODY END>

Example of submit_sm in form of byte array for CLIENTID:

Field	Size (Octets)	Type	Description
Parameter Tag	2	Integer	clientID
Length	2	Integer	Length of value part in octets
Value	15	Octet String	clientId attribute of card that is integer type with 15 digits length

```
0, 0, 0, -125, 0, 0, 0, 4, 0, 0, 0, 0, 92, -31, 45, -76, 0, 5, 0, 49, 0, 0, 0,
54, 49, 53, 53, 52, 52, 49, 49, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 70, 65, 99, 116,
105, 111, 110, 47, 82, 101, 103, 105, 115, 116, 114, 97, 116, 105, 111, 110, 32,
118, 105, 97, 32, 77, 73, 65, 8, 13, 10, 49, 48, 32, 84, 111, 107, 101, 110,
115, 32, 115, 101, 110, 116, 13, 10, 65, 58, 55, 48, 52, 50, 56, 54, 13, 10,
101, 110, 100, 32, 111, 102, 32, 115, 109, 115, 20, 0, 0, 15, 49, 50, 51, 52,
53, 54, 55, 56, 57, 48, 49, 50, 51, 52, 53
```

the TLV optional part will start from index=112

112 - 114 > [20, 0] > tag =5120

114-116 > [0,15] > len=15

116-131 > [49, 50, 51, 52, 53, 54, 55, 56, 57, 48, 49, 50, 51, 52, 53] > clintId=123456789012345

Example of submit_sm in form of HEX dump for CLIENTID:

```
00000083:00000004:00000000:5CE12DB4:
00050031:00000036:31353534:34313100:
00000000:00000000:00464163:74697661:
74696F6E:2F526567:69737472:6174696F:
6E207669:61204D49:413A0D0A:31302054:
6F6B656E:73207365:6E740D0A:413A3730:
34323836:0D0A656E:64206F66:20736D73:
1400000F:31323334:35363738:39303132:
333435
```

The TLV optional part will be:

1400000F:31323334:35363738:39303132:333435

1400 > client id tag > 5120 or 0x1400

000F > len of tag value > 15

31323334:35363738:39303132:333435 > clientid attribute value > 123456789012345

Message Security

The SMPP protocol does not enforce any security measures on its messaging; however, as the MQ server is not a real SMSC, **SMS via JMS Library** can encrypt the whole submit_sm message using an agreed 3DES key.

This option will not be available by default and the customer would need to officially request it before GPayments can commence the development.

Logging

The **SMS via JMS library** reads logging configuration from log4j.xml under the AA_HOME directory. The library logs messages and events in ERROR, INFO and DEBUG levels only.

Deployment

The **SMS via JMS library** will be embedded into ActiveAccess installation package as a library.

Installation

To install the **SMS via JMS library**, proceed with the following instructions.

- If ActiveAccess version is 6.8.0 or lower, upgrade it to v6.8.1 or later.
- Open AA_HOME/sms_jms_config.properties and update it with the preferred settings.
- Define a new SMS Centre in **MIA > System Management > Device Management > Edit Default Device Parameter-SMS > SMS Centre > New SMS Centre**

Test Scenario

Message Validation Failed

SMS via JMS library successfully connects to MQ Server but there are persisting issues during processing and validating message:

- Message processing failed
- Invalid Submit_SM parameters.

Message processing failed

Process on message failed due to expected or unexpected exceptions, such as UnsupportedEncodingException when converting to other message formats.

Invalid Submit_SM parameters

All parameters' length must be checked and if limitation is exceeded, an exception will be thrown. Other validation on sumbit_sm message is mentioned below:

- **source_addr:** The sent source_addr's ton differs from source_addr_ton.
- **destination_addr:** The destination_addr is not defined or its ton differs from dest_addr_ton.
- **data_coding:** The sent data_coding is not defined or differs from short_message coding.
- **short_message:** short_message length is more than 254 bytes or differs from sm_length.
- **message_payload:** message_payload and short_message are both used simultaneously.
- **sm_length:** sm_length differs from short_message length or is not zero while message_payload has value.

Successful SMS sending

SMS via JMS library successfully connects to MQ Server and sends the message.

Decoupled Authentication Adapter

New page added.

Adapter Loading Process

The ActiveAccess ACS performs Decoupled Authentication challenges through Decoupled Authentication Adapters, which connect an existing Decoupled Authentication system with ActiveAccess. From 3-D Secure 2.2, when Decoupled Authentication is required for challenge, the ACS triggers the external Decoupled Authentication process and performs interactions with the cardholder through Decoupled Authentication Adapters. In other words, the ACS communicates with the existing `DecoupledAuth-System` via a middle-ware, known as the Decoupled Authentication Adapter. The Decoupled Authentication Adapter can either be loaded locally by the ACS, or communicated via HTTP calls, known as the Native API and REST API versions respectively.

Native API Version of Decoupled Authentication Adapter

The Native API version of Decoupled Authentication Adapter is known as `decoupledAuth.adapter` in this specification. The Native Adapters are in the form of `JAR` files. Only Java is supported for the Native API version of Decoupled Authentication Adapter.

Native Decoupled Authentication Adapter developers provide the adapters in one or more JAR files. The `decoupledAuth.adapter` implementation in the adapter must implement the Java interface `Adapter` which is located in the `com.gpayments.decoupled.api.v1` package.

Implementation Steps

The steps to implement a Decoupled Authentication Adapter are as follows:

1. Create a Java project and obtain the corresponding Adapter API library from the ActiveAccess package. This library contains the interface definition for `decoupledAuth.adapter` (Native Decoupled Authentication Adapter). The library must be imported in the created project.

2. The Native API Adapter should be implemented as a service, based on the Decoupled Authentication interface `Adapter`. This specific implementation of the service is known as *service provider*. The `ACS` loads this class in startup and uses it in Decoupled Authentication. As a requirement of the `ACS`, the *provider class* must have a public zero-argument constructor so that it can be instantiated during loading.
3. A service provider is identified by placing a *provider-configuration file* in the resource directory `META-INF/services`. The file's name is the fully-qualified binary name of the service's type, e.g. `com.gpayments.decoupled.api.v1.Adapter`. The file contains a list of fully-qualified binary names of concrete provider classes, one per line. Space and tab characters surrounding each name, as well as blank lines, are ignored. The comment character is '`#`' ('\u0023', NUMBER SIGN); on each line all characters following the first comment character are ignored. The file must be encoded in `UTF-8`.

Adapter Interface Methods

The `Adapter` interface has four methods as follows:

- **Method Name:** *`getAdapterInfo`*
  - **Description:** Returns information about `decoupledAuth.adapter`
  - **Input:** This method takes no arguments
  - **Output:** An instance of `AdapterInfo` class that contains information about `decoupledAuth.adapter` should be returned. The `AdapterInfo` is explained in detail in the AdapterInfo Data Elements section.
- **Method Name:** *`ping`*
  - **Description:** Checks whether Decoupled-Authenticator Server is accessible or not
  - **Input:** This method takes no arguments
  - **Output:** The result of *Decoupled Authentication Server* pinging should be returned in a *boolean* value. If the server responds to *ping* successfully, `true` should be returned, otherwise `false` should be returned.
- **Method Name:** *`requestChallenge`*
  - **Description:** To call when Decoupled Authentication is necessary and returns whether authentication method is available for the card or not.

- **Input:** This method takes the following parameters:
  - **Field Name:** `acsTransactionId`
    - **Description:** Universally Unique identifier assigned by the `ACS` to identify a single transaction
    - **Length:** 36 characters
    - **Format:** String
    - **Accepted Value:** Canonical format as defined in IETF RFC 4122
    - **Message Inclusion:** Required
  - **Field Name:** `transactionInfo`
    - **Description:** Information about the transaction, which can be necessary for the Decoupled Authentication
    - **Format:** An instance of type `TransactionInfo`. `TransactionInfo` is explained in detail in the TransactionInfo Data Elements section.
    - **Message Inclusion:** Required
- **Output:** Adapter `requestChallenge` method has a `DecoupledRequestChallengeResult` return type
  - **Field Name:** `DecoupledRequestChallengeResult`
    - **Format:** An instance of type `DecoupledRequestChallengeResult`. `DecoupledRequestChallengeResult` is explained in detail in the DecoupledRequestChallengeResult Data Elements section.
    - **Message Inclusion:** Required

- **Method Name:** *`getChallengeResult`*
  - **Description:** Checks if the card is authenticated successfully or not
  - **Input:** This method takes the following parameters:
    - **Field Name:** `acsTransactionId`
      - **Description:** Universally Unique identifier assigned by the `ACS` to identify a single transaction
      - **Length:** 36 characters
      - **Format:** String
      - **Accepted Value:** Canonical format as defined in IETF RFC 4122

- **Message Inclusion:** Required

- **Field Name:** `decoupledTransId`
  - **Description:** Unique identifier assigned by the `Decoupled-Authenticator-System` to identify a single Decoupled Authentication Challenge
  - **Length:** Variable, maximum 36 characters
  - **Format:** String
  - **Message Inclusion:** Optional
- **Field Name:** `additionalInfo`
  - **Description:** Some additional Information for Decoupled Authentication
  - **Format:** JSON object of `AdditionalInfo` type. `AdditionalInfo` is explained in the AdditionalInfo Data Elements section.
  - **Message Inclusion:** Optional

- **Output:** Adapter `getChallengeResult` method has a `DecoupledAuthenticationResult` return type.
  - **Field Name:** `DecoupledAuthenticationResult`
    - **Format:** An instance of type `DecoupledAuthenticationResult`. `DecoupledAuthenticationResult` is explained in detail in the DecoupledAuthenticationResult Data Elements section.
    - **Message Inclusion:** Required

RESTful API version of Decoupled Adapter

For the Restful API version of the `decoupledAuth.adapter`, a Restful API needs to be defined similar to the adapter interface. `ACS` implements the Restful client, and the `Decoupled Auth API Server` will be implemented by the client. In this case, no JARs or plugins need to be loaded by `ACS`. Clients must provide a specific URL for `ACS`, known as `Adapter-URL` in this document, and the required REST API endpoints are defined based on this URL.

> **Note**
>
> There is also a Swagger API available for the sample REST Decoupled Adapter server and it is included in the ActiveAccess release package. To use it, run Decoupled Server in the installation package and then open `https://localhost:8448/swagger-ui.html#/` in your browser.

Get Decoupled Adapter Information

The ACS sends an HTTP request to get `decoupledAuth.adapter` information. The details of this request are:

- **URL:** `Adapter-URL` /adapter-info
- **Request Method:** GET
- **Request Parameters:** There are not any request parameters for this REST API
- **Response:**
  - **Name:** adapterInfo
  - **Format:** JSON object of `AdapterInfo`
  - **Description:** For further details on `AdapterInfo`, refer to the AdapterInfo Data Elements section
  - **Inclusion:** Required

EXAMPLE SAMPLE REQUEST

```
HTTP URL: https://localhost:8448/restful-adapter/decoupled/adapter-info
```

EXAMPLE SAMPLE RESPONSE

```
{
   "id":"4740fa3f-cd6c-40d8-b795-b9768f2f4c95",
   "name":"restful-adapter",
   "version":"1.0.0",
   "signature":"SIGNATURE",
   "maxAuthenticationTime":900
}
```

Check Decoupled Authenticator Server Availability Status

The ACS sends an HTTP request to check whether Decoupled Authenticator Server is accessible or not. The details of this request are:

- **URL:** `Adapter-URL` /ping
- **Request Method:** GET
- **Request Parameters:** There are not any request parameters for this REST API
- **Response:** If Adapter Server is accessible to perform the Decoupled Authentication challenge process, this API returns an HTTP entity with 200 (OK) HTTP status code.

Request Decoupled Authentication Challenge

ACS calls an HTTP API when Decoupled Authentication is necessary and returns whether authentication method for the card is available or not.

- **URL:** `Adapter-URL` /request-challenge/{acsTransactionId}
- **Request Method:** POST
- **Path variables:**
  - **Name:** acsTransactionId
  - **Format:** String
  - **Inclusion:** Required
- **Request Body:**
  - **Name:** transactionInfo
  - **Format:** JSON object of TransactionInfo type. For further details on TransactionInfo, refer to the TransactionInfo Data Elements section.
  - **Inclusion:** Required
- **Response:**
  - **Name:** DecoupledRequestChallengeResult
  - **Format:** JSON object of DecoupledRequestChallengeResult
  - **Description:** `DecoupledRequestChallengeResult` is explained in detail in the DecoupledRequestChallengeResult Data Elements section
  - **Inclusion:** Required

EXAMPLE SAMPLE REQUEST

```
    HTTP URL: http://localhost:8447/restful-adapter/decoupled/request-challenge/
da3cb8f9-90a2-489b-a7af-28ba33ce924a

    Body: {
      "acctNumber": 4548812049400004,
      "additionalInfo": {
        "callbackUrl": "string",
        "clientId": "123456789012345",
        "deviceId": "123e4567-e89b-12d3-a456-426655440000"
      },
      "cardHolderInfo": {
        "cardholderName": "cardholderName",
        "email": "abc@example.com",
        "homePhone": {
          "cc": 61,
          "subscriber": 234567890
        },
        "mobilePhone": {
          "cc": 61,
          "subscriber": 234567890
        },
        "shipAddrCity": "shipAddrCity",
        "shipAddrCountry": "040",
        "shipAddrLine1": "shipAddrLine1",
        "shipAddrLine2": "shipAddrLine2",
        "shipAddrLine3": "shipAddrLine3",
        "shipAddrPostCode": "shipAddrPostCode",
        "shipAddrState": "VIC",
        "workPhone": {
          "cc": 61,
          "subscriber": 234567890
        }
      },
      "deviceChannel": "01",
      "issuerName": "AnyBank",
      "merchantName": "merchantName",
      "messageCategory": "01",
      "purchaseAmount": 12345,
      "purchaseCurrency": "036",
      "purchaseExponent": 2,
      "threeDSServerTransID": "a4edc97f-4b89-4e52-8590-6c328f0b9648"
    }
```

EXAMPLE SAMPLE RESPONSE

```
    {
      "requestChallengeEnum": "OK",
```

```
    "decoupledTransId": "0679cb73-ea9a-41fb-8fda-dec78a46cd0b",
    "message": "message example"
}
```

Get Decoupled Authentication Result

`ACS` calls an HTTP API to check if the cardholder is authenticated successfully or not. The details of this REST API are:

- **URL:** `Adapter-URL` /challenge-result/{acsTransactionId}/{decoupledTransId} (or `Adapter-URL` /challenge-result/{acsTransactionId} if `decoupledTransId` is null.)
- **Request Method:** POST
- **Path Variables:**
  - **Name:** acsTransactionId
  - **Format:** String
  - **Inclusion:** Required
  - **Name:** decoupledTransId
  - **Format:** String
  - **Inclusion:** Optional
- **Request Body:**
  - **Name:** additionalInfo
  - **Format:** JSON object of `AdditionalInfo` type. You can find details of `AdditionalInfo` in the AdditionalInfo Data Elements section.
- **Response:**
  - **Name:** DecoupledAuthenticationResult
  - **Format:** JSON object of type `DecoupledAuthenticationResult`
  - **Description:** `DecoupledAuthenticationResult` is explained in detail in the DecoupledAuthenticationResult Data Elements section.
  - **Inclusion:** Required

Authentication Mechanism For the RESTful API Version

Certificate-based mutual authentication is used as the authentication mechanism for the RESTful API version. The steps are:

- The `ACS` publishes a CA for adapter communication, known as `Adapter CA`
- The `ACS` also issues a server certificate for the adapter
- The `ACS` uses a generated client certificate that is issued by the same `Adapter CA`
- The Adapter server implementation must be set up with the CA and mutual authentication provided
- The `ACS` will try to connect to the Adapter Server. If the connection can be established, then the `ACS` will continue with the adapter, otherwise it throws an error.

Adapter Data Elements

AdapterInfo Data Elements

- **Field Name:** `id`
  - **Description:** The `ACS` assigned *UUID* to the `Decoupled Adapter`
  - **Length:** 36 characters
  - **Format:** String
  - **Accepted Value:** Canonical format as defined in IETF RFC 4122
  - **Message Inclusion:** Required
- **Field Name:** `name`
  - **Description:** The `ACS` assigned *name* to the `Decoupled Adapter`
  - **Length:** Variable, maximum 100 characters
  - **Format:** String
  - **Accepted Value:**
  - **Message Inclusion:** Required
- **Field Name:** `version`
  - **Description:** The number that indicates the `Decoupled Adapter` version

- **Length:** Variable
- **Format:** String
- **Accepted Value:**
- **Message Inclusion:** Required

- **Field Name:** `signature`
  - **Description:** Signature to validate `Decoupled Adapter` integrity
  - **Length:** Variable
  - **Format:** String
  - **Accepted Value:**
  - **Message Inclusion:** Optional

> **Note**
>
> This field is not currently used and is reserved for future versions.

- **Field Name:** `maxAuthenticationTime`
  - **Description:** Maximum time in minutes to wait for decoupled server to complete its authentication
  - **Length:** Variable
  - **Format:** Integer
  - **Accepted Value:**
  - **Message Inclusion:** Required

TransactionInfo Data Elements

- **Field Name**: `threeDSServerTransID`
  - **Description:** Universally unique transaction identifier assigned by the 3DS Server to identify a single transaction
  - **Length:** 36 characters
  - **Format:** String

- **Accepted Value:** Canonical format as defined in IETF RFC 4122. May utilise any of the specified versions if the output meets specified requirements.
- **Device Channel:** 01-APP, 02-BRW, 03-3RI
- **Message Category:** 01-PA, 02-NPA
- **Message Inclusion:** Required

- **Field Name:** `additionalInfo`
  - **Description:** Some additional Info for Decoupled Authentication
  - **Length:**
  - **Format:** JSON object of `AdditionalInfo` type. `AdditionalInfo` is explained in the AdditionalInfo Data Elements section.
  - **Accepted Value:**
  - **Device Channel:** 01-APP, 02-BRW, 03-3RI
  - **Message Category:** 01-PA, 02-NPA
  - **Message Inclusion:** Optional
- **Field Name**: `purchaseAmount`
  - **Description:** Purchase amount in minor units of currency with all punctuation removed. When used in conjunction with the Purchase Currency Exponent field, proper punctuation can be calculated.
  - **Length:** 48 characters
  - **Format:** String
  - **Accepted Value:** Example: If the purchase amount is USD 123.45, the element will contain the value 12345.
  - **Device Channel:** 01-APP, 02-BRW
  - **Message Category:** 01-PA, 02-NPA
  - **Message Inclusion:** 01-PA: Required, 02-NPA: Conditional
  - **Conditional Inclusion:** Required for 02NPA if 3DS Requestor Authentication Indicator = 02 or 03.
- **Field Name**: `purchaseCurrency`
  - **Description:** Currency in which purchase amount is expressed.
  - **Length:** 3 characters

- **Format:** String
- **Accepted Value:** ISO 4217 three-digit currency code; 955-964 and 999 values are excluded and not permitted.
- **Device Channel:** 01-APP, 02-BRW
- **Message Category:** 01-PA, 02-NPA
- **Message Inclusion:** 01-PA: Required, 02-NPA: Conditional
- **Conditional Inclusion:** Required for 02NPA if 3DS Requestor Authentication Indicator = 02 or 03.

- **Field Name**: `purchaseExponent`
  - **Description:** Minor units of currency as specified in the ISO 4217 currency exponent.
  - **Length:** 1 character
  - **Format:** String
  - **Accepted Value:** Number
  - **Device Channel:** 01-APP, 02-BRW
  - **Message Category:** 01-PA, 02-NPA
  - **Message Inclusion:** 01-PA: Required, 02-NPA: Conditional
  - **Conditional Inclusion:** Required for 02NPA if 3DS Requestor Authentication Indicator = 02 or 03.
- **Field Name**: `messageCategory`
  - **Description:** Identifies the category of the message for a specific use case
  - **Length:** 2 characters
  - **Format:** String
  - **Accepted Value:** 01-PA, 02-NPA
  - **Device Channel:** 01-APP, 02-BRW, 03-3RI
  - **Message Category:** 01-PA, 02-NPA
  - **Message Inclusion:** Required
- **Field Name**: `purchaseDate`
  - **Description:** Date and time of the purchase, expressed in UTC timezone.
  - **Length:** 14 characters

- **Format:** String (Date Format: YYYYMMDDHHMMSS)
- **Accepted Value:**
- **Device Channel:** 01-APP, 02-BRW, 03-3RI
- **Message Category:** 01-PA, 02-NPA
- **Message Inclusion:** 1-PA: Required, 02-NPA: Conditional
- **Conditional Inclusion:** Conditional (Required if purchaseAmount is set)

- **Field Name**: `deviceChannel`
  - **Description:** Indicates the type of channel interface being used to initiate the transaction.
  - **Length:** 2 characters
  - **Format:** String
  - **Accepted Value:** 01 (APP); 02 (BRW); 03 (3RI)
  - **Device Channel:** 01-APP, 02-BRW, 03-3RI
  - **Message Category:** 01-PA, 02-NPA
  - **Message Inclusion:** Required
- **Field Name**: `acctNumber`
  - **Description:** Account number that will be used in the authorisation request for payment transactions. It will be represented by PAN, token.
  - **Length:** 13-19 characters, or up to 72 characters when encryption for sensitive data is enabled
  - **Format:** String
  - **Accepted Value:** Format represented ISO 7812.
  - **Device Channel:** 01-APP, 02-BRW, 03-3RI
  - **Message Category:** 01-PA, 02-NPA
  - **Message Inclusion:** Required
- **Field Name**: `merchantName`
  - **Description:** Merchant name assigned by the Acquirer or Payment System.
  - **Length:** Variable, maximum 40 characters
  - **Format:** String

- **Accepted Value:** Same name used in the authorization message as defined in ISO 8583.
- **Device Channel:** 01-APP, 02-BRW, 03-3RI
- **Message Category:** 01-PA, 02-NPA
- **Message Inclusion:** 01-PA: Required, 02-NPA: Optional
- **Conditional Inclusion:** Optional but strongly recommended to include for 02NPA if the merchant is also the 3DS Requestor.

- **Field Name**: `cardHolderInfo`
  - **Description:** Information about the Cardholder, which is provided by the 3DS Requestor. For further details, refer to the CardHolderInfo Data Elements section
  - **Type:** CardHolderInfo
- **Field Name**: `issuerName`
  - **Description:** Name of the Issuer
  - **Length:** Variable, maximum 64 characters.
  - **Format:** String
  - **Accepted Value:** Any
  - **Device Channel:** 01-APP, 02-BRW, 03-3RI
  - **Message Category:** 01-PA, 02-NPA
  - **Message Inclusion:** Required

AdditionalInfo Data Elements

- **Field Name:** `clientId`
  - **Description:** Client ID
  - **Length:** 15 characters
  - **Format:** String
  - **Accepted Value:** Decimal numbers
  - **Message Inclusion:** Optional (this field will be excluded from RESTful JSON message when it has no value)
- **Field Name:** `deviceId`
  - **Description:** Device ID

- **Length:** Variable, maximum 36 characters
- **Format:** String
- **Accepted Value:** Any
- **Message Inclusion:** Optional (this field will be excluded from RESTful JSON message when it has no value)

- **Field Name:** `callbackUrl`
  - **Description:** URL to be called by `DecoupledAuth system` when Decoupled Authentication result is prepared. ACS will call `getChallengeResult` after receiving this request.
  - **Length:** Variable, maximum 2048 characters
  - **Format:** String
  - **Accepted Value:** Fully qualified URL
  - **Message Inclusion:** Required

CardHolderInfo Data Elements

- **Field Name**: `cardholderName`
  - **Description:** Cardholder name
  - **Length:** 2-45 characters or up to 104 characters when encryption for sensitive data is enabled
  - **Format:** String
  - **Accepted Value:** Alphanumeric special characters, listed in EMVBook 4, “Appendix B”.
  - **Device Channel:** 01-APP, 02-BRW, 03-3RI
  - **Message Category:** 01-PA, 02-NPA
  - **Message Inclusion:** Conditional
  - **Conditional Inclusion:** Required unless market or regional mandate restricts sending this information.
- **Field Name**: `email`
  - **Description:** The email address associated with the account that is either entered by the Cardholder, or is on file with the 3DS Requestor.
  - **Length:** 254 characters

- **Format:** String
- **Accepted Value:** Shall meet requirements of Section 3.4 of IETF RFC 5322
- **Device Channel:** 01-APP, 02-BRW, 03-3RI
- **Message Category:** 01-PA, 02-NPA
- **Message Inclusion:** Conditional
- **Conditional Inclusion:** Required unless market or regional mandate restricts sending this information.

- **Field Name**: `homePhone`
  - **Description:** The home phone number provided by the cardholder. For further details, refer to the HomePhone Data Elements section
  - **Type:** HomePhone
  - **Device Channel:** 01-APP, 02-BRW, 03-3RI
  - **Message Category:** 01-PA, 02-NPA
  - **Message Inclusion:** Conditional
  - **Conditional Inclusion:** Required unless market or regional mandate restricts sending this information.
- **Field Name**: `mobilePhone`
  - **Description:** The mobile phone number provided by the cardholder. For further details, refer to the MobilePhone Data Elements section
  - **Type:** MobilePhone
  - **Device Channel:** 01-APP, 02-BRW, 03-3RI
  - **Message Category:** 01-PA, 02-NPA
  - **Message Inclusion:** Conditional
  - **Conditional Inclusion:** Required unless market or regional mandate restricts sending this information.
- **Field Name**: `shipAddrCity`
  - **Description:** City portion of the shipping address requested by the cardholder
  - **Length:** Variable, maximum 50 characters
  - **Format:** String
  - **Accepted Value:**

- **Device Channel:** 01-APP, 02-BRW, 03-3RI
- **Message Category:** 01-PA, 02-NPA
- **Message Inclusion:** Optional

- **Field Name**: `shipAddrCountry`
  - **Description:** Country of the shipping address requested by the cardholder.
  - **Length:** 3 characters
  - **Format:** String
  - **Accepted Value:** ISO 3166-1 three-digit country code; 901-999 values are excluded and not permitted.
  - **Device Channel:** 01-APP, 02-BRW, 03-3RI
  - **Message Category:** 01-PA, 02-NPA
  - **Message Inclusion:** Optional
- **Field Name**: `shipAddrLine1`
  - **Description:** First line of the street address or equivalent local portion of the shipping address requested by the cardholder
  - **Length:** Variable, maximum 50 characters
  - **Format:** String
  - **Accepted Value:**
  - **Device Channel:** 01-APP, 02-BRW, 03-3RI
  - **Message Category:** 01-PA, 02-NPA
  - **Message Inclusion:** Optional
- **Field Name**: `shipAddrLine2`
  - **Description:** Second line of the street address or equivalent local portion of the shipping address requested by the cardholder
  - **Length:** Variable, maximum 50 characters
  - **Format:** String
  - **Accepted Value:**
  - **Device Channel:** 01-APP, 02-BRW, 03-3RI
  - **Message Category:** 01-PA, 02-NPA

- **Message Inclusion:** Optional

- **Field Name**: `shipAddrLine3`
  - **Description:** Third line of the street address or equivalent local portion of the shipping address requested by the cardholder
  - **Length:** Variable, maximum 50 characters
  - **Format:** String
  - **Accepted Value:**
  - **Device Channel:** 01-APP, 02-BRW, 03-3RI
  - **Message Category:** 01-PA, 02-NPA
  - **Message Inclusion:** Optional
- **Field Name**: `shipAddrPostCode`
  - **Description:** The ZIP or other postal code of the shipping address requested by the cardholder
  - **Length:** Variable, maximum 16 characters
  - **Format:** String
  - **Accepted Value:**
  - **Device Channel:** 01-APP, 02-BRW, 03-3RI
  - **Message Category:** 01-PA, 02-NPA
  - **Message Inclusion:** Optional
- **Field Name**: `shipAddrState`
  - **Description:** The state or province of the shipping address associated with the card being used for this purchase
  - **Length:** Variable, maximum 3 characters
  - **Format:** String
  - **Accepted Value:** Should be the country subdivision code defined in ISO 3166-2.
  - **Device Channel:** 01-APP, 02-BRW, 03-3RI
  - **Message Category:** 01-PA, 02-NPA
  - **Message Inclusion:** Optional

- **Field Name**: `workPhone`
  - **Description:** The work phone number provided by the cardholder. For further details, refer to the WorkPhone Data Elements section
  - **Type:** WorkPhone
  - **Device Channel:** 01-APP, 02-BRW, 03-3RI
  - **Message Category:** 01-PA, 02-NPA
  - **Message Inclusion:** Conditional
  - **Conditional Inclusion:** Required (if available) unless market or regional mandate restricts sending this information.

HomePhone Data Elements

- **Field Name**: `cc`
  - **Description:** Country Code of the number
  - **Length:** 1-3 characters
  - **Format:** String
  - **Accepted Value:** Refer to ITU-E.164 for additional information on format and length
  - **Device Channel:** 01-APP, 02-BRW, 03-3RI
  - **Message Category:** 01-PA, 02-NPA
- **Field Name**: `subscriber`
  - **Description:** Subscriber sections of the number
  - **Length:** Variable, maximum 15 characters
  - **Format:** String
  - **Accepted Value:** Refer to ITU-E.164 for additional information on format and length
  - **Device Channel:** 01-APP, 02-BRW, 03-3RI
  - **Message Category:** 01-PA, 02-NPA

MobilePhone Data Elements

- **Field Name**: `cc`
  - **Description:** Country Code of the number
  - **Length:** 1-3 characters
  - **Format:** String
  - **Accepted Value:** Refer to ITU-E.164 for additional information on format and length
  - **Device Channel:** 01-APP, 02-BRW, 03-3RI
  - **Message Category:** 01-PA, 02-NPA
- **Field Name**: `subscriber`
  - **Description:** Subscriber sections of the number
  - **Length:** Variable, maximum 15 characters
  - **Format:** String
  - **Accepted Value:** Refer to ITU-E.164 for additional information on format and length
  - **Device Channel:** 01-APP, 02-BRW, 03-3RI
  - **Message Category:** 01-PA, 02-NPA

WorkPhone Data Elements

- **Field Name**: `cc`
  - **Description:** Country Code of the number
  - **Length:** 1-3 characters
  - **Format:** String
  - **Accepted Value:** Refer to ITU-E.164 for additional information on format and length
  - **Device Channel:** 01-APP, 02-BRW, 03-3RI
  - **Message Category:** 01-PA, 02-NPA
- **Field Name**: `subscriber`
  - **Description:** Subscriber sections of the number
  - **Length:** Variable, maximum 15 characters

- **Format:** String
- **Accepted Value:** Refer to ITU-E.164 for additional information on format and length
- **Device Channel:** 01-APP, 02-BRW, 03-3RI
- **Message Category:** 01-PA, 02-NPA

DecoupledRequestChallengeResult Data Elements

- **Field Name**: `requestChallengeResultEnum`
  - **Description:** Result of requesting Decoupled Authentication challenge to `Decoupled-Authenticator-System`, that determines whether Decoupled Authentication method is available for this card or not
  - **Length:**
  - **Format:**
  - **Accepted Value:** OK, ERROR, PROGRESS, TIMEOUT
  - **Accepted Value:**
  - **Message Inclusion:** Required
- **Field Name:** `decoupledTransId`
  - **Description:** Unique identifier assigned by the `Decoupled-Authenticator-System` to identify a single Decoupled Authentication Challenge
  - **Length:** Variable, maximum 36 characters
  - **Format:** String
  - **Accepted Value:**
  - **Message Inclusion:** Optional
- **Field Name**: `message`
  - **Description:** Any required message that should be returned to the `ACS`
  - **Length:** Variable, maximum 500 characters
  - **Format:** String
  - **Accepted Value:**
  - **Message Inclusion:** Optional

DecoupledAuthenticationResult Data Elements

- **Field Name**: `DecoupledResult`
  - **Description:** Result of Decoupled Authentication challenge
  - **Length:**
  - **Format:**
  - **Accepted Value:** AUTHENTICATED, NOT_AUTHENTICATED, ERROR
  - **Message Inclusion:** Required
- **Field Name**: `message`
  - **Description:** Any required message that should be returned to the `ACS`
  - **Length:** Variable, maximum 500 characters
  - **Format:** String
  - **Accepted Value:**
  - **Message Inclusion:** Optional

Out of Band Authentication Adapter

Adapter Loading Process

The ActiveAccess ACS performs Out Of Band (OOB) challenges through OOB Adapters, which connect the existing OOB authentication system with ActiveAccess. During 3DSecure 2.0 challenge flows where OOB Authentication is required, the `ACS` will trigger the external OOB process and perform interactions with the Cardholder via the OOB Adapters.

For this purpose, the `ACS` communicates with the existing `OOB-System` via a middleware, known as the OOB Adapter. The OOB Adapter can be either loaded locally by the ACS or communicated via HTTP calls, known as the Native API and REST API versions respectively.

Native API version of OOB Adapter

The Native API version of OOB Adapter is known as `oob.adapter` in this specification. The Native Adapters are provided in the form of `JAR` files, by GPayments or ActiveAccess clients. Only Java is supported for the Native API version of OOB Adapter.

Native OOB Adapter developers provide the adapters in one or more JAR files. The `oob.adapter` implementation in the adapter must implement the Java interface `Adapter` in the `com.gpayments.oob.api.v1` package.

Implementation Steps

The steps for implementing OOB Adapter are:

1. OOB Adapter developers create a Java project and obtain the corresponding Adapter API library from the ActiveAccess package. This library contains the interface definition for `oob.adapter` (Native OOB Adapter).
2. The Native API Adapter should be implemented as a Service based on the OOB interface `Adapter`. This specific implementation of the service known as *service provider*. `ACS` loads this class in startup and uses it in OOB Authentication. The requirement enforced by `ACS` is that *provider class* must have a public zero-argument constructor so that it can be instantiated during loading.

3. A service provider is identified by placing a *provider-configuration file* in the resource directory `META-INF/services`. The file's name is the fully-qualified binary name of the service's type, e.g. `com.gpayments.oob.api.v1.Adapter`. The file contains a list of fully-qualified binary names of concrete provider classes, one per line. Space and tab characters surrounding each name, as well as blank lines, are ignored. The comment character is '`#`' ('\u0023', NUMBER SIGN); on each line all characters following the first comment character are ignored. The file must be encoded in `UTF-8`.

Adapter Interface Methods

The `Adapter` interface has four methods as follows:

- **Method Name:** *`getAdapterInfo`*
  - **Description:** Returns information about `oob.adapter`
  - **Input:** This method takes no arguments
  - **Output:** An instance of `AdapterInfo` class that contains information about `oob.adapter` should be returned. The `AdapterInfo` is explained in detail in the AdapterInfo Data Elements section.
- **Method Name:** *`ping`*
  - **Description:** Checks whether OOB-Authenticator Server is accessible or not
  - **Input:** This method takes no arguments
  - **Output:** The result of *OOB Server* pinging should be returned in a *boolean* value. If the server responds to *ping* successfully, `true` should be returned, otherwise `false` should be returned.
- **Method Name:** *`requestChallenge`*
  - **Description:** To call when OOB authentication is necessary and returns whether authentication method for the card is available or not.
  - **Input:** This method takes the following parameters:
    - **Field Name:** `acsTransactionId`
      - **Description:** Universally Unique identifier assigned by the `ACS` to identify a single transaction
      - **Length:** 36 characters
      - **Format:** String

- **Accepted Value:** Canonical format as defined in IETF RFC 4122
- **Message Inclusion:** Required

- **Field Name:** `transactionInfo`
  - **Description:** Information about the transaction, which is required for the OOB authentication
  - **Length:**
  - **Format:** An instance of type `TransactionInfo`. `TransactionInfo` is explained in detail in the TransactionInfo Data Elements section.
  - **Accepted Value:**
  - **Message Inclusion:** Required

- **Output:** Adapter `requestChallenge` method has a `OobRequestChallengeResult` return type
  - **Field Name:** `oobRequestChallengeResult`
    - **Description:**
    - **Length:**
    - **Format:** An instance of type `OobRequestChallengeResult`. `OobRequestChallengeResult` is explained in details in the OobRequestChallengeResult Data Elements section.
    - **Accepted Value:**
    - **Message Inclusion:** Required

- **Method Name:** *`getChallengeResult`*
  - **Description:** Checks if the card is authenticated successfully or not
  - **Input:** This method takes the following parameters:
    - **Field Name:** `acsTransactionId`
      - **Description:** Universally Unique identifier assigned by the `ACS` to identify a single transaction.
      - **Length:** 36 characters
      - **Format:** String
      - **Accepted Value:** Canonical format as defined in IETF RFC 4122
      - **Message Inclusion:** Required

- **Field Name:** `oobTransId`
  - **Description:** Unique identifier assigned by the `OOB-Authenticator-System` to identify a single OOB Authentication Challenge
  - **Length:** Variable, maximum 36 characters.
  - **Format:** String
  - **Accepted Value:**
  - **Message Inclusion:** Optional
- **Field Name:** `additionalInfo`
  - **Description:** Some additional Information for OOB authentication
  - **Length:**
  - **Format:** JSON object of `AdditionalInfo` type. `AdditionalInfo` is explained in the AdditionalInfo Data Elements section.
  - **Accepted Value:**
  - **Message Inclusion:** Optional

- **Output:** Adapter `getChallengeResult` method has a `OobAuthenticationResult` return type.
  - **Field Name:** `oobAuthenticationResult`
    - **Description:**
    - **Length:**
    - **Format:** An instance of type `OobAuthenticationResult`. `OobAuthenticationResult` is explained in details in the OobAuthenticationResult Data Elements section.
    - **Accepted Value:**
    - **Message Inclusion:** Required

RESTful API version of OOB Adapter

For the Restful API version of the `oob.adapter`, a Restful API needs to be defined similar to the adapter interface. `ACS` implements the Restful client and the `OOB Adapter API Server` will be implemented by the client. In this case, no JARs or plugins need to be loaded by `ACS`. Clients

must provide a specific URL for `ACS`. Known as `Adapter-URL` in this document and the required REST API endpoints are defined based on this URL.

> **Note**
>
> We also provide a Swagger API for the sample REST OOB Adapter server that is included in release package. To use it, run OOB Server in the installation package and then open `https://localhost:8447/swagger-ui.html#/` in your browser.

Get OOB Adapter Information

The ACS sends an HTTP request to get `oob.adapter` information. The details of this request are:

- **URL:** `Adapter-URL` /adapter-info
- **Request Method:** GET
- **Request Parameters:** there are not any request parameters for this REST API
- **Response:**
  - **Name:** adapterInfo
  - **Format:** JSON object of `AdapterInfo`.
  - **Description:** For further details on `AdapterInfo`, refer to the AdapterInfo Data Elements section.
  - **Inclusion:** Required

Sample request

```
HTTP URL: http://localhost:8447/restful-adapter/oob/adapter-info
```

Sample response

```
{
  "id": "0b99b82f-62cf-4275-88b3-de039020f14e",
  "name": "restful-adapter",
  "version": "1",
  "signature": "SIGNATURE"
}
```

Check OOB Authenticator Server availability status

The ACS sends an HTTP request to check whether OOB Authenticator Server is accessible or not. The details of this request are:

- **URL:** `Adapter-URL` /ping
- **Request Method:** GET
- **Request Parameters:** there are not any request parameters for this REST API
- **Response:** If Adapter Server is accessible to perform the OOB challenge process, this API returns an HTTP entity with 200 (OK) HTTP status code.

Request OOB Challenge

ACS calls an HTTP API when OOB authentication is necessary and returns whether authentication method for the card is available or not.

- **URL:** `Adapter-URL` /request-challenge/{acsTransactionId}
- **Request Method:** POST
- **Path variables:**
  - **Name:** acsTransactionId
  - **Format:** String
  - **Inclusion:** required
- **Request Body:**
  - **Name:** transactionInfo
  - **Format:** JSON object of TransactionInfo type. For further details on TransactionInfo, refer to the TransactionInfo Data Elements section.
  - **Inclusion:** required
- **Response:**
  - **Name:** oobRequestChallengeResult
  - **Format:** JSON object of OobRequestChallengeResult.
  - **Description:** `OobRequestChallengeResult` is explained in details in the OobRequestChallengeResult Data Elements section.
  - **Inclusion:** required

Sample Request

```
HTTP URL: http://localhost:8447/restful-adapter/oob/request-challenge/da3cb8f9-90a2-489b-a7af-28ba33ce924a

Body: {
  "acctNumber": 4548812049400004,
  "additionalInfo": {
    "clientId": "123456789012345",
    "deviceId": "123e4567-e89b-12d3-a456-426655440000"
  },
  "cardHolderInfo": {
    "cardholderName": "cardholderName",
    "email": "abc@example.com",
    "homePhone": {
      "cc": 61,
      "subscriber": 234567890
    },
    "mobilePhone": {
      "cc": 61,
      "subscriber": 234567890
    },
    "shipAddrCity": "shipAddrCity",
    "shipAddrCountry": "040",
    "shipAddrLine1": "shipAddrLine1",
    "shipAddrLine2": "shipAddrLine2",
    "shipAddrLine3": "shipAddrLine3",
    "shipAddrPostCode": "shipAddrPostCode",
    "shipAddrState": "VIC",
    "workPhone": {
      "cc": 61,
      "subscriber": 234567890
    }
  },
  "deviceChannel": "01",
  "issuerName": "AnyBank",
  "merchantName": "merchantName",
  "messageCategory": "01",
  "purchaseAmount": 12345,
  "purchaseCurrency": "036",
  "purchaseExponent": 2,
  "threeDSServerTransID": "a4edc97f-4b89-4e52-8590-6c328f0b9648"
}
```

Sample response

```
{
  "requestChallengeEnum": "OK",
  "oobTransId": "0679cb73-ea9a-41fb-8fda-dec78a46cd0b",
  "message": null
}
```

Get OOB authentication result

`ACS` calls an HTTP API to check if the cardholder is authenticated successfully or not. The details of this REST API are:

- **URL:** `Adapter-URL` /challenge-result/{acsTransactionId}/{oobTransId} (or `Adapter-URL` / challenge-result/{acsTransactionId} if `oobTransId` is null.)
- **Request Method:** POST
- **Path variables:**
  - **Name:** acsTransactionId
  - **Format:** String
  - **Inclusion:** Required
  - **Name:** oobTransId
  - **Format:** String
  - **Inclusion:** Optional
- **Request Body:**
  - **Name:** additionalInfo
  - **Format:** JSON object of `AdditionalInfo` type. You can find details of `AdditionalInfo` in the AdditionalInfo Data Elements section.
- **Response:**
  - **Name:** oobAuthenticationResult
  - **Format:** JSON object of type `OobAuthenticationResult`
  - **Description:** `OobAuthenticationResult` is explained in details in the OobAuthenticationResult Data Elements section.
  - **Inclusion:** Required

Authentication mechanism for the RESTful API version

Certificate based mutual authentication is used as the authentication mechanism for the RESTful API version. The steps are:

- `ACS` publishes a CA for adapter communication, known as `Adapter CA`
- `ACS` also issues a server certificate for the adapter.

- `ACS` uses a generated client certificate that is issued by the same `Adapter CA`.
- The Adapter server implementation must be set up with the CA and mutual authentication provided.
- `ACS` will try to connect to the Adapter Server and if the connection can be established then `ACS` will continue with the adapter, otherwise it throws an error.

Adapter Data Elements

AdapterInfo Data Elements

- **Field Name:** `id`
  - **Description:** The `ACS` assigned *UUID* to the `OOB Adapter`
  - **Length:** 36 characters
  - **Format:** String
  - **Accepted Value:** Canonical format as defined in IETF RFC 4122
  - **Message Inclusion:** Required
- **Field Name:** `name`
  - **Description:** The `ACS` assigned *name* to the `OOB Adapter`
  - **Length:** Variable, maximum 100 characters
  - **Format:** String
  - **Accepted Value:**
  - **Message Inclusion:** Required
- **Field Name:** `version`
  - **Description:** The number that indicates the `OOB Adapter` version
  - **Length:** Variable
  - **Format:** String
  - **Accepted Value:**
  - **Message Inclusion:** Required
- **Field Name:** `signature`

> **Note**
>
> This field is not currently used and is reserved for for future versions.

- **Description:** Signature to validate `OOB Adapter` integrity
- **Length:** Variable
- **Format:** String
- **Accepted Value:**
- **Message Inclusion:** Optional

TransactionInfo Data Elements

- **Field Name**: `threeDSServerTransID`
  - **Description:** Universally unique transaction identifier assigned by the 3DS Server to identify a single transaction.
  - **Length:** 36 characters
  - **Format:** String
  - **Accepted Value:** Canonical format as defined in IETF RFC 4122. May utilise any of the specified versions if the output meets specified requirements.
  - **Device Channel:** 01-APP, 02-BRW, 03-3RI
  - **Message Category:** 01-PA, 02-NPA
  - **Message Inclusion:** Required
- **Field Name:** `additionalInfo`
  - **Description:** Some additional Info for OOB authentication
  - **Length:**
  - **Format:** JSON object of `AdditionalInfo` type. `AdditionalInfo` is explained in the AdditionalInfo Data Elements section.
  - **Accepted Value:**
  - **Device Channel:** 01-APP, 02-BRW, 03-3RI
  - **Message Category:** 01-PA, 02-NPA
  - **Message Inclusion:** Optional

- **Field Name**: `purchaseAmount`
  - **Description:** Purchase amount in minor units of currency with all punctuation removed. When used in conjunction with the Purchase Currency Exponent field, proper punctuation can be calculated.
  - **Length:** 48 characters
  - **Format:** String
  - **Accepted Value:** Example: If the purchase amount is USD 123.45, element will contain the value 12345.
  - **Device Channel:** 01-APP, 02-BRW
  - **Message Category:** 01-PA, 02-NPA
  - **Message Inclusion:** 01-PA: Required, 02-NPA: Conditional
  - **Conditional Inclusion:** Required for 02NPA if 3DS Requestor Authentication Indicator = 02 or 03.
- **Field Name**: `purchaseCurrency`
  - **Description:** Currency in which purchase amount is expressed.
  - **Length:** 3 characters
  - **Format:** String
  - **Accepted Value:** ISO 4217 three-digit currency code; 955-964 and 999 values are excluded and not permitted.
  - **Device Channel:** 01-APP, 02-BRW
  - **Message Category:** 01-PA, 02-NPA
  - **Message Inclusion:** 01-PA: Required, 02-NPA: Conditional
  - **Conditional Inclusion:** Required for 02NPA if 3DS Requestor Authentication Indicator = 02 or 03.
- **Field Name**: `purchaseExponent`
  - **Description:** Minor units of currency as specified in the ISO 4217 currency exponent.
  - **Length:** 1 character
  - **Format:** String
  - **Accepted Value:** Number
  - **Device Channel:** 01-APP, 02-BRW

- **Message Category:** 01-PA, 02-NPA
- **Message Inclusion:** 01-PA: Required, 02-NPA: Conditional
- **Conditional Inclusion:** Required for 02NPA if 3DS Requestor Authentication Indicator = 02 or 03.

- **Field Name**: `messageCategory`
  - **Description:** Identifies the category of the message for a specific use case
  - **Length:** 2 characters
  - **Format:** String
  - **Accepted Value:** 01-PA, 02-NPA
  - **Device Channel:** 01-APP, 02-BRW, 03-3RI
  - **Message Category:** 01-PA, 02-NPA
  - **Message Inclusion:** Required
- **Field Name**: `purchaseDate`
  - **Description:** Date and time of the purchase, expressed in UTC timezone.
  - **Length:** 14 characters
  - **Format:** String (Date Format: YYYYMMDDHHMMSS)
  - **Accepted Value:**
  - **Device Channel:** 01-APP, 02-BRW, 03-3RI
  - **Message Category:** 01-PA, 02-NPA
  - **Message Inclusion:** 1-PA: Required, 02-NPA: Conditional
  - **Conditional Inclusion:** Conditional (Required if purchaseAmount is set)
- **Field Name**: `deviceChannel`
  - **Description:** Indicates the type of channel interface being used to initiate the transaction.
  - **Length:** 2 characters
  - **Format:** String
  - **Accepted Value:** 01 (APP); 02 (BRW); 03 (3RI)
  - **Device Channel:** 01-APP, 02-BRW, 03-3RI
  - **Message Category:** 01-PA, 02-NPA

- **Message Inclusion:** Required

- **Field Name**: `acctNumber`
  - **Description:** Account number that will be used in the authorisation request for payment transactions. It will be represented by PAN, token.
  - **Length:** 13-19 characters, or up to 72 characters when encryption for sensitive data is enabled
  - **Format:** String
  - **Accepted Value:** Format represented ISO 7812.
  - **Device Channel:** 01-APP, 02-BRW, 03-3RI
  - **Message Category:** 01-PA, 02-NPA
  - **Message Inclusion:** Required
- **Field Name**: `merchantName`
  - **Description:** Merchant name assigned by the Acquirer or Payment System.
  - **Length:** Variable, maximum 40 characters
  - **Format:** String
  - **Accepted Value:** Same name used in the authorisation message as defined in ISO 8583.
  - **Device Channel:** 01-APP, 02-BRW, 03-3RI
  - **Message Category:** 01-PA, 02-NPA
  - **Message Inclusion:** 01-PA: Required, 02-NPA: Optional
  - **Conditional Inclusion:** Optional but strongly recommended to include for 02NPA if the merchant is also the 3DS Requestor.
- **Field Name**: `cardHolderInfo`
  - **Description:** Information about the Cardholder, which is provided by the 3DS Requestor. For further details, refer to the CardHolderInfo Data Elements section
  - **Type:** CardHolderInfo
- **Field Name**: `issuerName`
  - **Description:** Name of the Issuer
  - **Length:** Variable, maximum 64 characters.
  - **Format:** String
  - **Accepted Value:** Any.

- **Device Channel:** 01-APP, 02-BRW, 03-3RI
- **Message Category:** 01-PA, 02-NPA
- **Message Inclusion:** Required

- ➕ **Field Name**: `threeDSRequestorAppURL`
  - **Description:** 3DS Requestor App declaring their URL within the CReq message so that the Authentication app can call the 3DS Requestor App after OOB authentication has occurred. Each transaction would require a unique Transaction ID by using the SDK Transaction ID.
  - **Length:** Variable, maximum 256 characters
  - **Format:** String
  - **Accepted Value:** Fully qualified URL
  - **Message Inclusion:** Optional

AdditionalInfo Data Elements

- **Field Name:** `clientId`
  - **Description:** Client ID
  - **Length:** 15 characters
  - **Format:** String
  - **Accepted Value:** Decimal numbers
  - **Message Inclusion:** Optional (this field will be excluded from RESTful JSON message when it has no value)
- **Field Name:** `deviceId`
  - **Description:** Device ID
  - **Length:** Variable, maximum 36 characters
  - **Format:** String
  - **Accepted Value:** Any
  - **Message Inclusion:** Optional (this field will be excluded from RESTful JSON message when it has no value)

- **Field Name:** `callbackUrl`
  - **Description:** URL to be called by `OOB system` when OOB authentication result is prepared. ACS will call `getChallengeResult` after receiving this request.
  - **Length:** Variable, maximum 2048 characters
  - **Format:** String
  - **Accepted Value:** Fully qualified URL
  - **Message Inclusion:** Required

CardHolderInfo Data Elements

- **Field Name**: `cardholderName`
  - **Description:** Cardholder name
  - **Length:** 2-45 characters or up to 104 characters when encryption for sensitive data is enabled
  - **Format:** String
  - **Accepted Value:** Alphanumeric special characters, listed in EMVBook 4, “Appendix B”.
  - **Device Channel:** 01-APP, 02-BRW, 03-3RI
  - **Message Category:** 01-PA, 02-NPA
  - **Message Inclusion:** Conditional
  - **Conditional Inclusion:** Required unless market or regional mandate restricts sending this information.
- **Field Name**: `email`
  - **Description:** The email address associated with the account that is either entered by the Cardholder, or is on file with the 3DS Requestor.
  - **Length:** 254 characters
  - **Format:** String
  - **Accepted Value:** Shall meet requirements of Section 3.4 of IETF RFC 5322.
  - **Device Channel:** 01-APP, 02-BRW, 03-3RI
  - **Message Category:** 01-PA, 02-NPA
  - **Message Inclusion:** Conditional

- **Conditional Inclusion:** Required unless market or regional mandate restricts sending this information.

- **Field Name**: `homePhone`
  - **Description:** The home phone number provided by the Cardholder. For further details, refer to the HomePhone Data Elements section
  - **Type:** HomePhone
  - **Device Channel:** 01-APP, 02-BRW, 03-3RI
  - **Message Category:** 01-PA, 02-NPA
  - **Message Inclusion:** Conditional
  - **Conditional Inclusion:** Required unless market or regional mandate restricts sending this information.
- **Field Name**: `mobilePhone`
  - **Description:** The mobile phone number provided by the Cardholder. For further details, refer to the MobilePhone Data Elements section
  - **Type:** MobilePhone
  - **Device Channel:** 01-APP, 02-BRW, 03-3RI
  - **Message Category:** 01-PA, 02-NPA
  - **Message Inclusion:** Conditional
  - **Conditional Inclusion:** Required unless market or regional mandate restricts sending this information.
- **Field Name**: `shipAddrCity`
  - **Description:** City portion of the shipping address requested by the Cardholder
  - **Length:** Variable, maximum 50 characters
  - **Format:** String
  - **Accepted Value:**
  - **Device Channel:** 01-APP, 02-BRW, 03-3RI
  - **Message Category:** 01-PA, 02-NPA
  - **Message Inclusion:** Optional
- **Field Name**: `shipAddrCountry`
  - **Description:** Country of the shipping address requested by the Cardholder.

- **Length:** 3 characters
- **Format:** String
- **Accepted Value:** ISO 3166-1 three-digit country code; 901-999 values are excluded and not permitted.
- **Device Channel:** 01-APP, 02-BRW, 03-3RI
- **Message Category:** 01-PA, 02-NPA
- **Message Inclusion:** Optional

• **Field Name**: `shipAddrLine1`

- **Description:** First line of the street address or equivalent local portion of the shipping address requested by the Cardholder.
- **Length:** Variable, maximum 50 characters
- **Format:** String
- **Accepted Value:**
- **Device Channel:** 01-APP, 02-BRW, 03-3RI
- **Message Category:** 01-PA, 02-NPA
- **Message Inclusion:** Optional

• **Field Name**: `shipAddrLine2`

- **Description:** Second line of the street address or equivalent local portion of the shipping address requested by the Cardholder.
- **Length:** Variable, maximum 50 characters
- **Format:** String
- **Accepted Value:**
- **Device Channel:** 01-APP, 02-BRW, 03-3RI
- **Message Category:** 01-PA, 02-NPA
- **Message Inclusion:** Optional

• **Field Name**: `shipAddrLine3`

- **Description:** Third line of the street address or equivalent local portion of the shipping address requested by the Cardholder.
- **Length:** Variable, maximum 50 characters
- **Format:** String

- **Accepted Value:**
- **Device Channel:** 01-APP, 02-BRW, 03-3RI
- **Message Category:** 01-PA, 02-NPA
- **Message Inclusion:** Optional

- **Field Name**: `shipAddrPostCode`
  - **Description:** The ZIP or other postal code of the shipping address requested by the Cardholder
  - **Length:** Variable, maximum 16 characters
  - **Format:** String
  - **Accepted Value:**
  - **Device Channel:** 01-APP, 02-BRW, 03-3RI
  - **Message Category:** 01-PA, 02-NPA
  - **Message Inclusion:** Optional
- **Field Name**: `shipAddrState`
  - **Description:** The state or province of the shipping address associated with the card being used for this purchase.
  - **Length:** Variable, maximum 3 characters
  - **Format:** String
  - **Accepted Value:** Should be the country subdivision code defined in ISO 3166-2.
  - **Device Channel:** 01-APP, 02-BRW, 03-3RI
  - **Message Category:** 01-PA, 02-NPA
  - **Message Inclusion:** Optional
- **Field Name**: `workPhone`
  - **Description:** The work phone number provided by the Cardholder. For further details, refer to the WorkPhone Data Elements section
  - **Type:** WorkPhone
  - **Device Channel:** 01-APP, 02-BRW, 03-3RI
  - **Message Category:** 01-PA, 02-NPA
  - **Message Inclusion:** Conditional

- **Conditional Inclusion:** Required (if available) unless market or regional mandate restricts sending this information.

HomePhone Data Elements

- **Field Name**: `cc`
  - **Description:** Country Code of the number
  - **Length:** 1-3 characters
  - **Format:** String
  - **Accepted Value:** Refer to ITU-E.164 for additional information on format and length.
  - **Device Channel:** 01-APP, 02-BRW, 03-3RI
  - **Message Category:** 01-PA, 02-NPA
- **Field Name**: `subscriber`
  - **Description:** Subscriber sections of the number
  - **Length:** Variable, maximum 15 characters
  - **Format:** String
  - **Accepted Value:** Refer to ITU-E.164 for additional information on format and length.
  - **Device Channel:** 01-APP, 02-BRW, 03-3RI
  - **Message Category:** 01-PA, 02-NPA

MobilePhone Data Elements

- **Field Name**: `cc`
  - **Description:** Country Code of the number
  - **Length:** 1-3 characters
  - **Format:** String
  - **Accepted Value:** Refer to ITU-E.164 for additional information on format and length.
  - **Device Channel:** 01-APP, 02-BRW, 03-3RI
  - **Message Category:** 01-PA, 02-NPA

- **Field Name**: `subscriber`
  - **Description:** Subscriber sections of the number
  - **Length:** Variable, maximum 15 characters
  - **Format:** String
  - **Accepted Value:** Refer to ITU-E.164 for additional information on format and length.
  - **Device Channel:** 01-APP, 02-BRW, 03-3RI
  - **Message Category:** 01-PA, 02-NPA

WorkPhone Data Elements

- **Field Name**: `cc`
  - **Description:** Country Code of the number
  - **Length:** 1-3 characters
  - **Format:** String
  - **Accepted Value:** Refer to ITU-E.164 for additional information on format and length.
  - **Device Channel:** 01-APP, 02-BRW, 03-3RI
  - **Message Category:** 01-PA, 02-NPA
- **Field Name**: `subscriber`
  - **Description:** Subscriber sections of the number
  - **Length:** Variable, maximum 15 characters
  - **Format:** String
  - **Accepted Value:** Refer to ITU-E.164 for additional information on format and length.
  - **Device Channel:** 01-APP, 02-BRW, 03-3RI
  - **Message Category:** 01-PA, 02-NPA

OobRequestChallengeResult Data Elements

- **Field Name**: `requestChallengeEnum`
  - **Description:** Result of requesting OOB authentication challenge to `OOB-Authenticator-System`, that determines whether OOB authentication method is available for this card or not.
  - **Length:**
  - **Format:**
  - **Accepted Value:** OK, ERROR
  - **Accepted Value:**
  - **Message Inclusion:** Required
- **Field Name:** `oobTransId`
  - **Description:** Unique identifier assigned by the `OOB-Authenticator-System` to identify a single OOB Authentication Challenge.
  - **Length:** Variable, maximum 36 characters
  - **Format:** String
  - **Accepted Value:**
  - **Message Inclusion:** Optional
- **Field Name**: `message`
  - **Description:** Any required message that should be returned to the `ACS`
  - **Length:** Variable, maximum 500 characters
  - **Format:** String
  - **Accepted Value:**
  - **Message Inclusion:** Optional
- **Field Name**: `instruction`
  - **Description:** Cardholder instructions on how to perform the OOB authentication
  - **Length:** Variable, maximum 350 characters
  - **Format:** String
  - **Accepted Value:** Any
  - **Message Inclusion:** Optional

OobAuthenticationResult Data Elements

- **Field Name**: `authenticationResultEnum`
  - **Description:** Result of OOB authentication challenge
  - **Length:**
  - **Format:**
  - **Accepted Value:** AUTHENTICATED, NOT_AUTHENTICATED, NOT_AUTHENTICATED_END, ERROR, PENDING
  - **Message Inclusion:** Required
- **Field Name**: `message`
  - **Description:** Any required message that should be returned to the `ACS`
  - **Length:** Variable, maximum 500 characters
  - **Format:** String
  - **Accepted Value:**
  - **Message Inclusion:** Optional
- ➕ **Field Name**: `instruction`
  - **Description:** Cardholder instructions on how to perform the OOB authentication
  - **Length:** Variable, maximum 350 characters
  - **Format:** String
  - **Accepted Value:** Any
  - **Message Inclusion:** Optional

Risk Engine Adapter

About Risk Adapter

When cardholder authentication occurs during a transaction, a challenge may be necessary because the transaction is deemed high-risk, e.g. above certain thresholds.

To assess the risk associated with the transaction and if a challenge to the cardholder is necessary, the `ACS` sends/receives proper data elements to/from risk assessment systems via middleware, known as `Risk Adapters`.

How RBA works

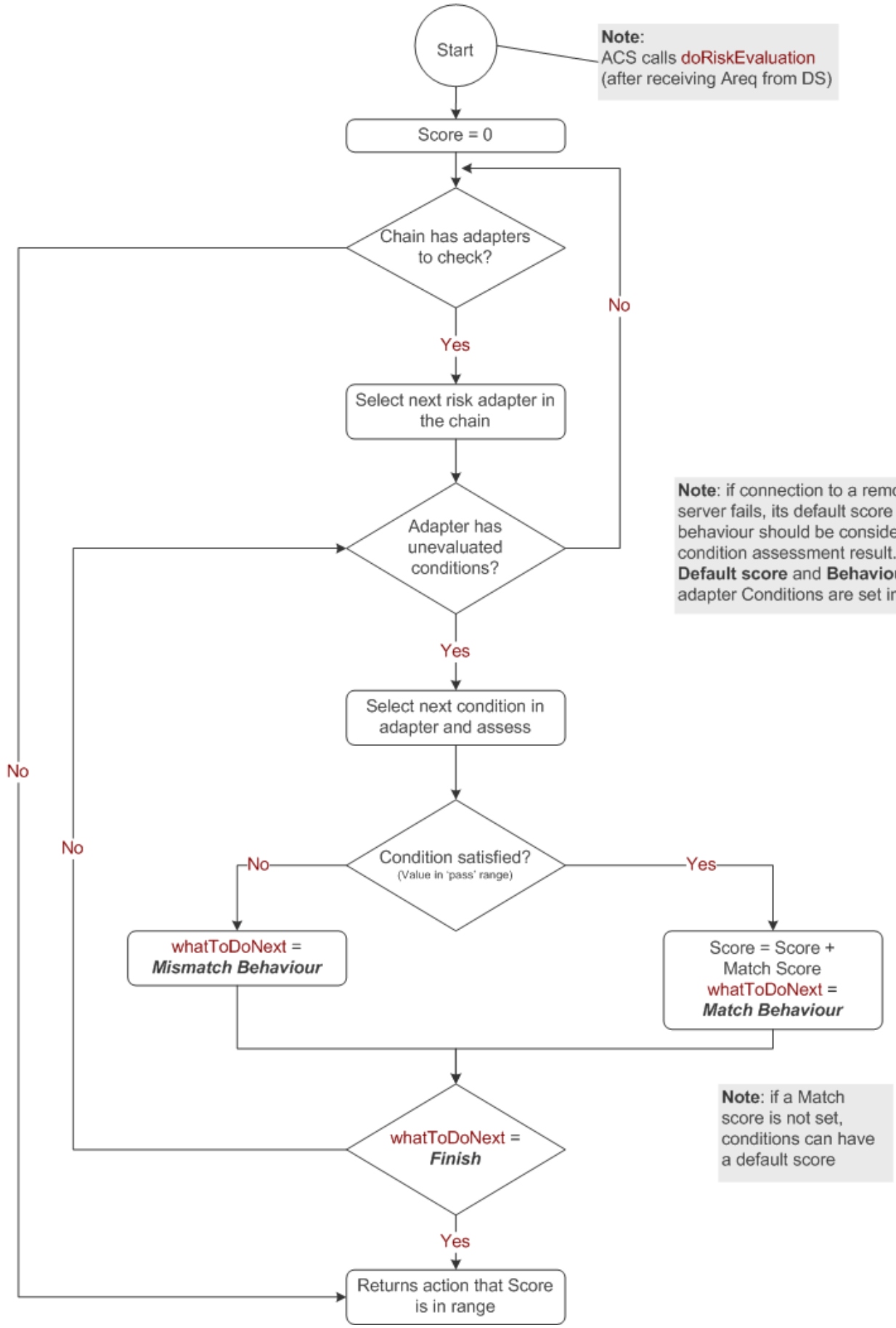
ActiveAccess has been developed to support risk through the following models:

- **Internal Risk**: Risk rules pre-defined internally within the application by GPayments. This may include samples or rules commonly used by issuers. The rules are configurable on the Administration UI.
- **External Risk via Adapter**: You can create your customised rules in the form of JAR files, as per the Risk Adapter Specification. Once the path of the JAR files has been included in the ActiveAccess configuration file (`<AA_HOME>/activeaccess.properties`), the rules will be configurable on the Administration UI in the same way as the Internal Risk model. Basic or complex rules can be defined based on the issuer's specific requirements
- **Remote Risk via Adapter**: Rules are defined on an external server and called by ActiveAccess using a URL configured on the Administration UI.

The RBA adapters are also designed to have a configurable scoring system. Scoring is fully configurable and each score range can be linked with an authentication method: frictionless, frictionless with review, static password, device or OOB.

The following diagram provides an overview of the risk evaluation process.

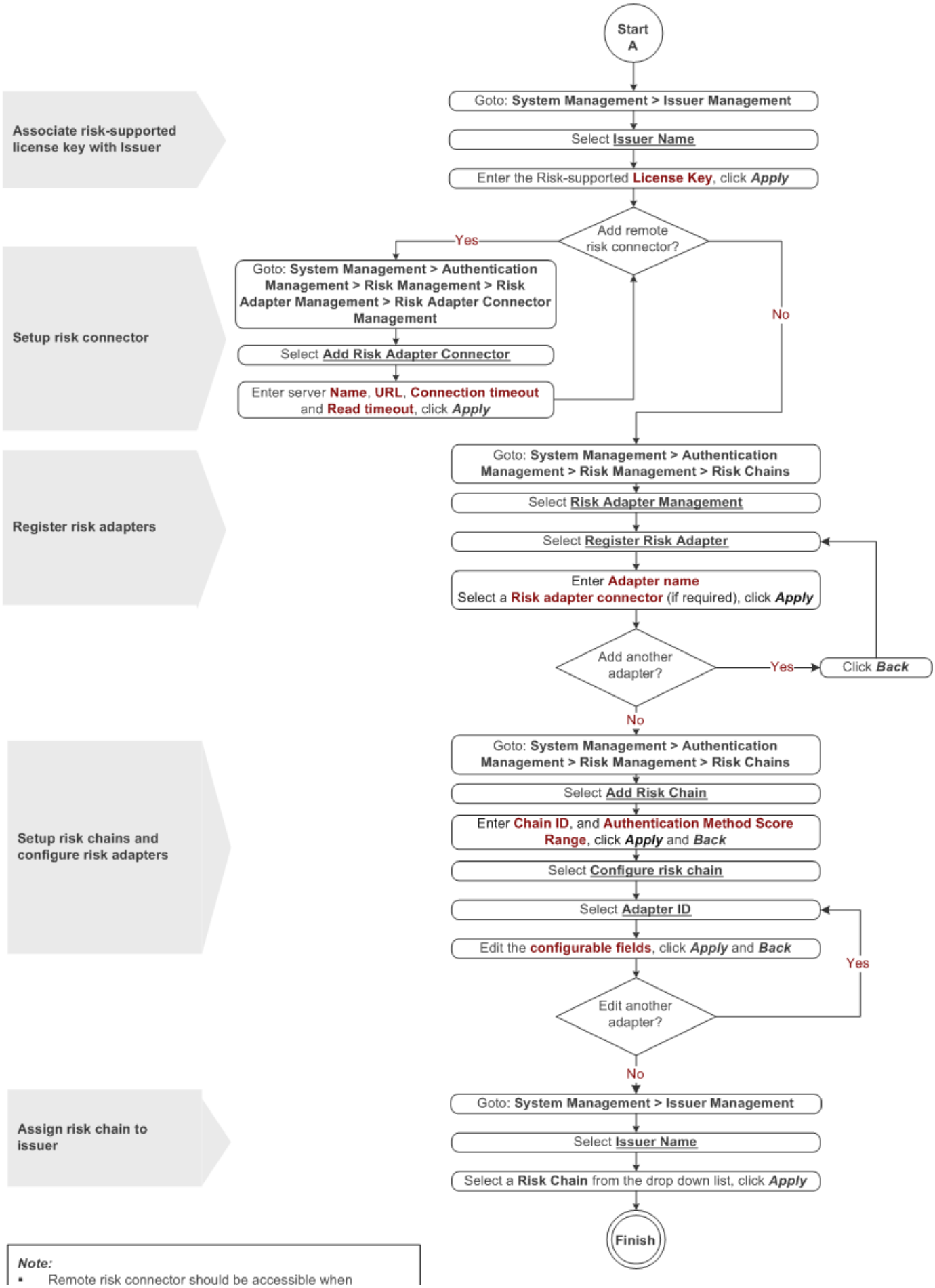
Risk Evaluation



Each `Risk Adapter` has one `Parameter` and one or more `Conditions`. Risk assessment systems perform tasks based on values received for these attributes. `Risk Adapters` can be either Native APIs or REST APIs. These Risk Adapter APIs are explained in further detail in the following sections.

The following diagram shows how risk chains are setup and risk adapters are registered on the Administration UI.

Risk chain setup



Note:

- Remote risk connector should be accessible when

Risk Adapter Specification

Native API Risk Adapter

The Native API Risk Adapter is a JAR file that implements an `ACS` specified adapter interface, known as `Adapter`, in a JAR file named `rba.adapter`. Note that you can rename your `Risk Adapters` as appropriate for your use.

Adapter development considerations

The following steps must be considered in `Risk Adapter` development:

1. Use the correct version of `risk.adapter-x` for your `ACS` version (where **x** represents the risk adapter API version) in your `Risk Adapter` project.
2. The `Adapter` interface in the `risk.adapter-x` JAR file must be implemented in your `Risk Adapter` project. This implementation is known as the *risk adapter provider*. The `ACS` loads this class at startup and uses it for transaction risk assessment. The `ACS` enforces the requirement for the *provider class* to have a public zero-argument constructor so that it can be instantiated during `ACS` loading.
3. A *risk adapter provider* is identified by placing a *provider-configuration file* in the resource directory `META-INF/services`. The file's name is the fully-qualified binary name of the service's type, e.g. `com.gpayments.rba.api.v1.Adapter`. The file contains a list of fully-qualified binary names of concrete provider classes, one per line. Space and tab characters surrounding each name, as well as blank lines, are ignored. The comment character is '`#`' ('\u0023', NUMBER SIGN); on each line all characters following the first comment character are ignored. The file must be encoded in `UTF-8`.

Adapter interface methods

The `Adapter` interface has four methods, which must be implemented in `Risk Adapter`:

- **Method Name:** `getAdapterId`
  - **Description:** Returns the `ACS` assigned *UUID* to `Risk Adapter`
  - **Input:** This method takes no arguments.
  - **Output:** The output of this method has the following properties:
    - **Description:** `Risk Adapter` ID.

- **Length:** 36 characters
- **Format:** String
- **Accepted Value:** Canonical format as defined in IETF RFC 4122
- **Message Inclusion:** Required

- **Method Name:** `getAdapterInfo`
  - **Description:** Returns some information about `Risk Adapter`
  - **Input:** This method takes no arguments
  - **Output:** An instance of `AdapterInfo` class that contains some information about `Risk Adapter` `AdapterInfo` is described in AdapterInfo Data Elements
- **Method Name:** `getParameter`
  - **Description:** Returns the used `Parameter` in the `Risk Adapter`.
  - **Input:** This method takes no arguments.
  - **Output:** An instance of `Parameter` class that contains some information about the used `Parameter` in Adapter conditions The `Parameter` class is Described in Parameter Data Elements
- **Method Name:** `getConditions`
  - **Description:** Returns a list of `Risk Adapter` conditions
  - **Input:** This method takes no arguments.
  - **Output:** List of available `Conditions` in `Risk Adapter`. The `Condition` class is described in Condition Data Elements

---

RESTful API Risk Adapter

For the RESTful API `Risk Adapter`, you should provide REST API endpoints that are accessible to the `ACS`. Note that `Adapter-URL` refers to the fully qualified RESTful `Risk Adapter` URL that will be used in transaction risk assessment.

> **Note**
>
> We also provide a Swagger API for the sample REST RBA server that is included in the release package. To use it, run RBA Server and open `https://localhost:8446/swagger-ui.html#/` in your browser.

Get Risk Adapter Information

The `ACS` sends an HTTP request to retrieve information about `Risk Adapter`, as follows:

- **URL:** `Adapter-URL`
- **Request Method:** GET
- **Request Parameters:** there is no request parameter for this REST API
- **Response:**
  - **Name:** restfulRBAAdapterInfo
  - **Format** JSON object of `RestfulRBAAdapterInfo` type.
  - **Description** `RestfulRBAAdapterInfo`. Described in RestfulRBAAdapterInfo Data Elements
  - **Inclusion:** Required

Transaction Risk Assessment

To estimate the risk for the transaction, the `ACS` calls an HTTP API of RESTful `Risk Adapter`, as follows:

- **URL:** `Adapter-URL`
- **Request Method:** POST
- **Request Body:**
  - **Name:** remoteAssessmentRequest
  - **Format:** JSON object of type `RemoteAssessmentRequest`, described in RemoteAssessmentRequest Data Elements
  - **Inclusion:** Required
- **Response:**
  - **Name:** assessmentResult
  - **Format:** JSON object of type `AssessmentResult`, described in AssessmentResult Data Elements
  - **Inclusion:** Required

Authentication mechanism for the RESTful API

Certificate based mutual authentication is used as the authentication mechanism for the RESTful API, using the following steps:

- `ACS` publishes a CA for adapter communication, in this instance named `Adapter CA`
- `ACS` also issues a server certificate for the adapter. The server certificate should have a serial number attribute set to the AA generated Adapter ID or something to identify the RESTful adapter.
- `ACS` uses a client generated certificate issued by the same `Adapter CA`.
- the Adapter server implementation must be setup with the CA provided and mutual authentication.
- `ACS` will try to connect to the Adapter Server and, if the connection can be established and the serial number matches the record in the database for this adapter, `ACS` will continue with the adapter otherwise it will throw an error.

AdapterInfo Data Elements

- **Field Name:** `id`
  - **Description:** The `ACS` assigned *UUID* to the `Risk Adapter`
  - **Length:** 36 characters
  - **Format:** String
  - **Accepted Value:** Canonical format as defined in IETF RFC 4122
  - **Message Inclusion:** Required
- **Field Name:** `name`
  - **Description:** The `ACS` assigned *name* to the `Risk Adapter`
  - **Length:** Variable, maximum 100 Characters
  - **Format:** String
  - **Accepted Value:**
  - **Message Inclusion:** Required

- **Field Name:** `version`
  - **Description:** The number that indicates the `Risk Adapter` version
  - **Length:** Variable
  - **Format:** String
  - **Accepted Value:**
  - **Message Inclusion:** Required
- **Field Name:** `signature`
  - **Description:** Signature to validate `Risk Adapter` integrity, currently not used, will be introduced in a future version
  - **Length:** Variable
  - **Format:** String
  - **Accepted Value:**
  - **Message Inclusion:** Optional

Parameter Data Elements

- **Field Name:** `name`
  - **Description:** The `Risk Adapter` *name* assigned to this parameter
  - **Length:** 50 characters
  - **Format:** String
  - **Accepted Value:** any
  - **Message Inclusion:** Required
- **Field Name:** `displayName`
  - **Description:** The `Risk Adapter` assigned *name* for displaying this parameter
  - **Length:** 50 characters
  - **Format:** String
  - **Accepted Value:** any
  - **Message Inclusion:** Required

- **Field Name:** `paramType`
  - **Description:** The type of Risk Adapter `Parameter`.
  - **Length:**
  - **Format:** ValueType. `ValueType` is described in ValueType Data Elements.
  - **Accepted Value:** NULL, NUMERIC, STRING, RANGE, LIST_OF_NUMERIC and LIST_OF_STRING
  - **Message Inclusion:** Required
- **Field Name:** `validator`
  - **Description:** A validator for this `Parameter`. This validator is an instance of a class that implements the `ParameterValidator` interface.
  - **Length:**
  - **Format:** An instance of `ParameterValidator` interface implementor. **This field should be ignored in mapping the `Parameter` object to the JSON object**
  - **Accepted Value:**
  - **Message Inclusion:**

The `ParameterValidator` interface has a method named `isParameterDataValid` and returns the *boolean* result of parameter validation. This method takes two arguments: * The first argument of this method, is an instance of type `AReq`, described in AReq Data Elements. * The second is an instance of a callback class that implements `TxCallback`. `TxCallback` is described in TxCallback Data Elements.

Condition Data Elements

- **Field Name:** `adapter`
  - **Description:** The `Adapter` that the current `Condition` belongs to; a reference to the current adapter
  - **Length:**
  - **Format:** An instance of type `Adapter`. **This field should be ignored in mapping `Condition` object to JSON object**
  - **Accepted Value:**

- **Message Inclusion:** Required

- **Field Name:** `boundParameter`
  - **Description:** The `Parameter` that is used in this adapter. The current `Condition` uses it in the risk assessment process; a reference to the current Risk Adapter Parameter
  - **Length:**
  - **Format:**
  - **Accepted Value:**
  - **Message Inclusion:** Required
- **Field Name:** `name`
  - **Description:** The `Risk Adapter` *name* assigned to this condition
  - **Length:** 50 characters
  - **Format:** String
  - **Accepted Value:** any
  - **Message Inclusion:** Required
- **Field Name:** `displayName`
  - **Description:** The `Risk Adapter` *name* assigned for displaying this condition
  - **Length:** 50 characters
  - **Format:** String
  - **Accepted Value:** any
  - **Message Inclusion:** Required
- **Field Name:** `valueType`
  - **Description:** The value types that should be set for `Condition` values
  - **Length:**
  - **Format:** `ValueType`
  - **Accepted Value:** NULL, NUMERIC, STRING, RANGE, LIST_OF_NUMERIC and LIST_OF_STRING
  - **Message Inclusion:** Required

- **Field Name:** `assessor`
  - **Description:** The condition assessor that is assigned to the adapter `Condition` and is responsible for getting the condition assessment result
  - **Length:**
  - **Format:** An instance of `ConditionAssessor` interface implementor. `ConditionAssessor` interface is described in ConditionAssessor Data Elements **This field should be ignored in mapping the `Condition` object to the JSON object.**
  - **Accepted Value:**
  - **Message Inclusion:** Required

ValueType Data Elements

`ValueType` is an enum type with six constants that represent the type of the value. These named constants are `NULL`, `NUMERIC`, `STRING`, `RANGE`, `LIST_OF_NUMERIC` and `LIST_OF_STRING`.

The `ValueType` class also has an instance method named `validateValue` that takes an instance of type `ConditionValue` as method arguments and validate if the value matches the required type. The `validateValue` does not return any information but throws NullPointerException or IllegalArgumentException when the input value is null or does not match with indicated valueType.

ConditionAssessor Data Elements

The interface `ConditionAssessor` has one method, named `assess`. This method takes five parameters of type `AReq`, `AdditionalInfo`, `TxCallback`, `Condition` and `ConditionValue` and returns the assessment result as `AssessmentResult`. `ConditionValue` and `AdditionalInfo` are described in ConditionValue Data Elements and AdditionalInfo Data Elements. `AssessmentResult` is described in AssessmentResult Data Elements.

TxCallback Data Elements

The `TxCallback` interface has callback mechanism so that ACS can provide proper historic transactions to the adapter. This interface has one method named `onPreviousTXRequiredByCountOrDays` that takes two integer arguments and returns a list of `AReqWithTransStatus` objects. The first parameter is the number of last transactions that should be returned and the second one is the number of days, which we should return the transactions occurred therein. The maximum transactions according to these parameter will be returned as result.

AssessmentResult Data Elements

- **Field Name:** `score`
  - **Description:** Risk assessment as a number
  - **Length:**
  - **Format:** Integer
  - **Accepted Value:** 0-100
  - **Message Inclusion:** Required
- **Field Name:** `whatToDoNext`
  - **Description:** The behavior that should be done about the remained risk-chain assessments
  - **Length:**
  - **Format:** JSON object of `behaviour` .
  - **Accepted Value:** CONTINUE, FINISH
  - **Message Inclusion:** Required

RestfulRBAAdapterInfo Data Elements

- **Field Name:** `adapterInfo`
  - **Description:** Some information about `Risk Adapter`

- **Length:**
- **Format:** JSON format of the `AdapterInfo` object, described in AdapterInfo Data Elements
- **Accepted Value:**
- **Message Inclusion:** Required

- **Field Name:** `parameter`
  - **Description:** The `Parameter` used in the `Risk Adapter`
  - **Length:**
  - **Format:** JSON format of the `Parameter` object, described in Parameter Data Elements
  - **Accepted Value:**
  - **Message Inclusion:** Required
- **Field Name:** `conditions`
  - **Description:** List of available conditions (`RemoteCondition`) in the current adapters
  - **Length:**
  - **Format:** List of `RemoteCondition` objects in JSON format, described in RemoteCondition Data Elements
  - **Accepted Value:**
  - **Message Inclusion:** Required

RemoteCondition Data Elements

In addition to the aforementioned `Condition` properties, `RemoteCondition` has two additional fields that may be necessary for historic data

- **Field Name:** `previousTx`
  - **Description:** Indicates the number of previous transactions required for risk assessment
  - **Length:**
  - **Format:** Integer
  - **Accepted Value:**
  - **Message Inclusion:** Optional

- **Field Name:** `previousTxInDays`
  - **Description:** Indicates the number of days of previous transactions required for risk assessment
  - **Length:**
  - **Format:** Integer
  - **Accepted Value:**
  - **Message Inclusion:** Optional

RemoteAssessmentRequest Data Elements

- **Field Name:** `aReq`
  - **Description:** The *AReq* message of the current transaction.
  - **Length:**
  - **Format:** JSON object of `AReq` type, described in AReq Data Elements
  - **Accepted Value:**
  - **Message Inclusion:** Required
- **Field Name:** `additionalInfo`
  - **Description:** Some additional Info for risk assessment
  - **Length:**
  - **Format:** JSON object of `AdditionalInfo` type. `AdditionalInfo` is explained in AdditionalInfo Data Elements section.
  - **Accepted Value:**
  - **Message Inclusion:** Optional
- **Field Name:** `previousData`
  - **Description:** The historic data required for risk assessment
  - **Length:**
  - **Format:** List of AReqWithTransStatus in JSON format. AReqWithTransStatus is explained in AReqWithTransStatus Data Elements section.
  - **Accepted Value:**

- **Message Inclusion:** Optional

- **Field Name:** `conditionName`
  - **Description:** Name of `Condition` to be processed
  - **Length:** 50 characters
  - **Format:** String
  - **Accepted Value:** Any
  - **Message Inclusion:** Required
- **Field Name:** `conditionValue`
  - **Description:** Some settings for `Condition`, such as condition value, default behaviour and the output score
  - **Length:**
  - **Format:** JSON object of `ConditionValue` type, described in ConditionValue Data Elements
  - **Accepted Value:**
  - **Message Inclusion:** Required

ConditionValue Data Elements

ConditionValue has the following private instance variables and some methods to access them.

- **Field Name:** `condition`
  - **Description:** `Condition` that the condition value is assigned to
  - **Length:**
  - **Format:** JSON object of `Condition`, described in Condition Data Elements
  - **Accepted Value:**
  - **Message Inclusion:** Required
- **Field Name:** `numeric`
  - **Description:** The condition value passed to the `Risk Adapter` if, and only if, `ValueType` of `Condition` is `NUMERIC`
  - **Length:**

- **Format:** Number
- **Accepted Value:**
- **Message Inclusion:** Required

- **Field Name:** `range`
  - **Description:** The condition value to be passed to the `Risk Adapter`, only if `ValueType` of `Condition` is equal to `RANGE` type
  - **Length:**
  - **Format:** An object of `Range` class. `Range` is explained in Range Data Elements
  - **Accepted Value:**
  - **Message Inclusion:** Required
- **Field Name:** `string`
  - **Description:** The condition value to be passed to the `Risk Adapter` if, and only if, `ValueType` of `Condition` is `STRING`
  - **Length:**
  - **Format:**
  - **Accepted Value:**
  - **Message Inclusion:** Required
- **Field Name:** `listOfNumeric`
  - **Description:** The condition value to be passed to the `Risk Adapter` if, and only if, `ValueType` of `Condition` is `LIST_OF_NUMERIC`
  - **Length:**
  - **Format:**
  - **Accepted Value:**
  - **Message Inclusion:** Required
- **Field Name:** `listOfString`
  - **Description:** The condition value to be passed to the `Risk Adapter` if, and only if, `ValueType` of `Condition` is `LIST_OF_STRING`
  - **Length:**
  - **Format:**
  - **Accepted Value:**

    - **Message Inclusion:** Required
- **Field Name:** `whenMatches`
    - **Description:** The default behaviour when the condition matches
    - **Length:**
    - **Format:** `behaviour`
    - **Accepted Value:** `CONTINUE` , `FINISH`
    - **Message Inclusion:** Required
- **Field Name:** `whenMismatch`
    - **Description:** The default behaviour when the condition mismatches
    - **Length:**
    - **Format:** `behaviour`
    - **Accepted Value:** `CONTINUE` , `FINISH`
    - **Message Inclusion:** Required
- **Field Name:** `scoreWhenMatches`
    - **Description:** The output score when the condition matches
    - **Length:**
    - **Format:** Integer
    - **Accepted Value:** 0-100
    - **Message Inclusion:** Required

Here is the methods that can be used in Native API Risk Adapter:

- `asNumeric(Condition condition, Number number)` is a static method that takes two arguments. The first one is an instance of `Condition` class and the second one is a value of type `Number` . By calling this method, a new bject of `ConditionValue` class will be created and initialize the field for `number` . Then this new object will be assigned to the condition (first argument) and will be returned as method output.
- `asString(Condition condition, String value)` is a static method that takes two arguments. The first one is an instance of `Condition` class and the second one is a value of type `String` . By calling this method, a new bject of `ConditionValue` class will be created and initialize the field for `string` . Then this new object will be assigned to the condition (first argument) and will be returned as method output.

- `asRange(Condition condition, Number start, Number end)` is a static method that takes three arguments. The first one is an instance of `Condition` class, the second one is a value of type `Number` as minimum value of a range and the third one is a value of type `Number` as maximum value of a range. By calling this method, a new object of `ConditionValue` class will be created and initialize the field for `range`. Then this new object will be assigned to the condition (first argument) and will be returned as method output.
- `asListOfNumeric(Condition condition, List<Number> value)` is a static method that takes two arguments. The first one is an instance of `Condition` class and the second one is a List of `Number`. By calling this method, a new object of `ConditionValue` class will be created and initialize the field for `listOfNumeric`. Then this new object will be assigned to the condition (first argument) and will be returned as method output.
- `asListOfString(Condition condition, List<String> value)` is a static method that takes two arguments. The first one is an instance of `Condition` class and the second one is a List of `String`. By calling this method, a new object of `ConditionValue` class will be created and initialize the field for `listOfString`. Then this new object will be assigned to the condition (first argument) and will be returned as method output.

Also there is getter and setter methods for retrieving and updating value of `ConditionValue` instance variables.

Range Data Elements

A class to represent ranges of values. A range is defined to contain all the values between the `start` and `end` values, where the start and end values are considered included in the range. This class has two instance variable of type `Number` that indicate minimum and maximum of the range.

- **Field Name:** `start`
  - **Description:** The minimum value of the range. This value included in the range.
  - **Length:**
  - **Format:** Number
  - **Accepted Value:**
  - **Message Inclusion:**

- **Field Name:** `end`
  - **Description:** The maximum value of the range. This value included in the range.
  - **Length:**
  - **Format:** Number
  - **Accepted Value:**
  - **Message Inclusion:**

The `Range` class also has two methods:

- `valueOf` is a static method that takes two number as start and end of the range and returns an instance of `Range` class.
- `inRange` is an instance method that takes a value and returns a boolean. This method checks whether the input value is in the specified range or not.

AReqWithTransStatus Data Elements

- **Field Name:** `aReq`
  - **Description:** The AReq message of the transaction.
  - **Length:**
  - **Format:** JSON object of AReq type. `AReq` is explained in AReq Data Elements section.
  - **Accepted Value:**
  - **Message Inclusion:** Required
- **Field Name:** `transStatus`
  - **Description:** Indicates whether a transaction qualifies as an authenticated transaction or account verification.
  - **Length:** 1 character
  - **Format:** String
  - **Accepted Value:**
    - Y = Authentication/ Account Verification Successful
    - N = Not Authenticated/ Account Not Verified; Transaction denied

        - U = Authentication/ Account Verification Could Not Be Performed; Technical or other problem, as indicated in ARes or RReq
        - A = Attempts Processing Performed; Not Authenticated/Verified, but a proof of attempted authentication/verification is provided
        - C = ChallengeRequired; Additional authentication is required using the CReq/CRes
        - R = Authentication/Account Verification Rejected; Issuer is rejecting authentication/ verification and request that authorisation not be attempted.
    - **Message Inclusion:** Optional
- **Field Name:** `transStatusReason`
    - **Description:** Provides information on why the Transaction Status field has the specified value.
    - **Length:** 2 characters
    - **Format:** String
    - **Accepted Value:**
        - 01 = Card authentication failed
        - 02 = Unknown Device
        - 03 = Unsupported Device
        - 04 = Exceeds authentication frequency limit
        - 05 = Expired card
        - 06 = Invalid card number
        - 07 = Invalid transaction
        - 08 = No Card record
        - 09 = Security failure
        - 10 = Stolen card
        - 11 = Suspected fraud
        - 12 = Transaction not permitted to cardholder
        - 13 = Cardholder not enrolled in service
        - 14 = Transaction timed out at the ACS
        - 15 = Low confidence
        - 16 = Medium confidence

    - 17 = High confidence
    - 18 = Very High confidence
    - 19 = Exceeds ACS maximum challenges
    - 20 = Non-Payment transaction not supported
    - 21 = 3RI transaction not supported
    - 22–79 = Reserved for EMVCo future use (values invalid until defined by EMVCo)
    - 80–99 = Reserved for DS use
  - **Message Inclusion:** Optional
- **Field Name:** `challenge`
  - **Description:** A value of `true` indicates that the transaction flow is Challenge.
  - **Length:**
  - **Format:** Boolean
  - **Accepted Value:** true, false
  - **Message Inclusion:** Optional
- **Field Name:** `challengeCancel`
  - **Description:** Indicator informing the ACS and the DS that the authentication has been cancelled.
  - **Length:** 2 characters
  - **Format:** String
  - **Accepted Value:**
  - 01 = No preference
  - 02 = No challenge requested
  - 03 = Challenge requested: 3DS Requestor Preference
  - 04 = Challenge requested: Mandate
  - 05–79 = Reserved for EMV future use (values invalid until defined by EMV 3DS)
  - 80-99 = Reserved for DS use
  - **Message Inclusion:** Optional
- **Field Name:** `errorCode`
  - **Description:** Code indicating the type of problem identified in the message.

- **Length:** 3 characters
- **Format:** String
- **Accepted Value:**
- **Message Inclusion:** Optional

AReq Data Elements

- **Field Name:** `threeDSCompInd`
  - **Description:** Indicates whether the 3DS Method was successfully completed
  - **Length:** 1 character
  - **Format:** String
  - **Accepted Value:**
    - Y = Successfully completed
    - N = Did not successfully complete
    - U = Unavailable
  - **Message Inclusion:** Optional
- **Field Name:** `threeDSRequestorAuthenticationInd`
  - **Description:** Indicates the type of authentication request. This data element provides additional information to the ACS to determine the best approach for handling the authentication request
  - **Length:** 2 characters
  - **Format:** String
  - **Accepted Value:**
    - 01 = Cardholder selected “Cancel”
    - 02 = 3DS Requestor cancelled Authentication.
    - 03 = Transaction Abandoned
    - 04 = Transaction Timed Out at ACS— other timeouts
    - 05 = Transaction Timed Out at ACS—First CReq not received by ACS
    - 06 = Transaction Error
    - 07 = Unknown

        - 08–79 = Reserved for future EMVCo use(values invalid until defined by EMVCo)
        - 80-99 = Reserved for future DS use
    - **Message Inclusion:** Optional
- **Field Name:** `threeDSRequestorAuthenticationInfo`
    - **Description:** Data that documents and supports a specific authentication process. In the current version of the specification, this data element is not defined in detail, however the intention is that for each 3DS requestor authentication method, this field will carry the data that the ACS will use to verify the authentication process, described in threeDSRequestorAuthenticationInfo Data Elements
    - **Length:** Variable
    - **Format:** Object
    - **Accepted Value:**
    - **Message Inclusion:** Optional
    - **Conditional Inclusion:** Optional but inclusion recommended
- **Field Name:** `threeDSRequestorChallengeInd`
    - **Description:** Indicates whether a challenge is to be requested for the transaction
    - **Length:** 2 characters
    - **Format:** String
    - **Accepted Value:**
        - 01 = No preference
        - 02 = No challenge requested
        - 03 = Challenge requested: 3DS Requestor Preference
        - 04 = Challenge requested: Mandate
        - 05–79 = Reserved for EMVCo future use (values invalid until defined by EMVCo)
        - 80-99 = Reserved for DS use
    - **Message Inclusion:** Optional
- **Field Name:** `threeDSRequestorID`
    - **Description:** DS assigned 3DS Requestor identifier.
    - **Length:** Variable, maximum 35 characters
    - **Format:** String

- **Accepted Value:** Any individual DS may impose specific formatting and character requirements on the contents of this field
- **Message Inclusion:** Required

- **Field Name:** `threeDSRequestorName`
  - **Description:** DS assigned 3DS Requestor name.
  - **Length:** Variable, maximum 40 characters
  - **Format:** String
  - **Accepted Value:** Any individual DS may impose specific formatting and character requirements on the contents of this field
  - **Message Inclusion:** Required
- **Field Name:** `threeDSRequestorPriorAuthenticationInfo`
  - **Description:** Information about how the 3DS Requestor authenticated the cardholder as part of a previous 3DS transaction, described in threeDSRequestorPriorAuthenticationInfo Data Elements
  - **Length:** Variable
  - **Format:** Object
  - **Accepted Value:**
  - **Message Inclusion:** Optional
  - **Conditional Inclusion:** Optional but inclusion recommended
- **Field Name:** `threeDSRequestorURL`
  - **Description:** Fully qualified URL of 3DS Requestor website or customer care site. This data element provides additional information to the receiving 3-D Secure system for when a problem arises, and should provide contact information.
  - **Length:** Variable, maximum 2048 characters
  - **Format:** String
  - **Accepted Value:** Fully qualified URL
  - **Message Inclusion:** Required
- **Field Name:** `threeDSServerRefNumber`
  - **Description:** Unique identifier assigned by the EMVCo Secretariat upon testing and approval

- **Length:** Variable, maximum 32 characters
- **Format:** String
- **Accepted Value:** Set by the EMVCo Secretariat
- **Message Inclusion:** Required

- **Field Name:** `threeDSServerOperatorID`
  - **Description:** DS assigned 3DS Server identifier. Each DS can provide a unique ID to each 3DS Server
  - **Length:** Variable, maximum 32 characters
  - **Format:** String
  - **Accepted Value:** Any individual DS may impose specific formatting and character requirements on the contents of this field
  - **Message Inclusion:** Conditional
  - **Conditional Inclusion:** Requirements for the presence of this field are DS specific
- **Field Name:** `threeDSServerTransID`
  - **Description:** Universally unique transaction identifier assigned by the 3DS Server to identify a single transaction
  - **Length:** 36 characters
  - **Format:** String
  - **Accepted Value:** Canonical format as defined in IETF RFC 4122. May utilise any of the specified versions if the output meets specified requirements
  - **Message Inclusion:** Required
- **Field Name:** `threeDSServerURL`
  - **Description:** Fully qualified URL of the 3DS Server to which the DS will send the RReq message after the challenge has completed. Incorrect formatting will result in a failure to deliver the transaction results via the RReq message.
  - **Length:** Variable, maximum 2048 characters
  - **Format:** String
  - **Accepted Value:** Fully qualified URL
  - **Message Inclusion:** Required

- **Field Name:** `threeRIInd`
  - **Description:** Indicates the type of 3RI request. This data element provides additional information to the ACS to determine the best approach for handling the 3RI request
  - **Length:** 2 characters
  - **Format:** String
  - **Accepted Value:**
    - 01 = Recurring transaction
    - 02 = Instalment transaction
    - 03 = Add card
    - 04 = Maintain card information
    - 05 = Account verification
    - 06–79 = Reserved for EMVCo future use (values invalid until defined by EMVCo)
    - 80-99 = Reserved for DS use
  - **Message Inclusion:** Required
- **Field Name:** `acctType`
  - **Description:** Indicates the type of account. For example, for a multi-account card product
  - **Length:** 2 characters
  - **Format:** String
  - **Accepted Value:**
    - 01 = Not Applicable
    - 02 = Credit
    - 03 = Debit
  - **Message Inclusion:** Conditional
  - **Conditional Inclusion:** Required if 3DS Requestor is asking cardholder which Account Type they are using before making the purchase Required in some markets (for example, for Merchants in Brazil). Otherwise, it is optional.
- **Field Name:** `acquirerBIN`
  - **Description:** Acquiring institution identification code as assigned by the DS receiving the AReq message

- **Length:** Variable, maximum 11 characters
- **Format:** String
- **Accepted Value:** This value correlates to the Acquirer BIN as defined by each Payment System or DS.
- **Message Inclusion:** 01-PA: Required, 02-NPA: Optional

- **Field Name:** `acquirerMerchantID`
  - **Description:** Acquirer-assigned Merchant identifier. This may be the same value that is used in authorisation requests sent on behalf of the 3DS Requestor and is represented in ISO 8583 formatting requirements
  - **Length:** Variable, maximum 35 characters
  - **Format:** String
  - **Accepted Value:** Individual DS may impose specific formatting and character requirements on the contents of this field
  - **Message Inclusion:** 01-PA: Required, 02-NPA: Optional
- **Field Name:** `addrMatch`
  - **Description:** Indicates whether the cardholder Shipping Address and cardholder Billing Address are the same
  - **Length:** 1 characters
  - **Format:** String
  - **Accepted Value:**
    - Y = Shipping Address matches Billing Address
    - N = Shipping Address does not match Billing Address
  - **Message Inclusion:** Optional
- **Field Name:** `broadInfo`
  - **Description:** Unstructured information sent between the 3DS Server, the DS and the ACS
  - **Length:** 4096
  - **Format:** String (JSON object)
  - **Accepted Value:**
  - **Message Inclusion:** Conditional
  - **Conditional Inclusion:** Requirements for the presence of this field are DS specific

- **Field Name:** `browserAcceptHeader`
  - **Description:** Exact content of the HTTP accept headers as sent to the 3DS Requestor from the cardholder's browser
  - **Length:** Variable, maximum 2048 characters
  - **Format:** String
  - **Accepted Value:** If the total length of the accept header sent by the browser exceeds 2048 characters, the 3DS Server truncates the excess portion
  - **Message Inclusion:** Required
- **Field Name:** `browserIP`
  - **Description:** IP address of the browser as returned by the HTTP headers to the 3DS Requestor
  - **Length:** Variable, maximum 45 characters
  - **Format:** String
  - **Accepted Value:**
    - IPv4 address is represented in the decimal format of 4 sets of decimal numbers separated by dots. The decimal number in each and every set is in the range 0 to 255. Example IPv4 address: 1.12.123.255
    - IPv6 address is represented as eight groups of four hexadecimal digits, each group representing 16 bits (two octets). The groups are separated by colons (:). Example IPv6 address: 2011:0db8:85 a3:0101:0101:8a2e:03 70:7334
  - **Message Inclusion:** Conditional
  - **Conditional Inclusion:** Include this field where regionally acceptable.
- **Field Name:** `browserJavaEnabled`
  - **Description:** Boolean that represents the ability of the cardholder browser to execute Java. Value is returned from the navigator. javaEnabled property.
  - **Length:**
  - **Format:** Boolean
  - **Accepted Value:** true, false
  - **Message Inclusion:** Required

- **Field Name:** `browserLanguage`
  - **Description:** Value representing the browser language as defined in IETF BCP47. Returned from navigator.language property
  - **Length:** Variable, 1-8 characters
  - **Format:** String
  - **Accepted Value:**
  - **Message Inclusion:** Required
- **Field Name:** `browserColorDepth`
  - **Description:** Value representing the bit depth of the colour palette for displaying images, in bits per pixel. Obtained from the cardholder's browser using the screen.colorDepth property.
  - **Length:** Variable, 1-2 characters
  - **Format:** String
  - **Accepted Value:**
    - 1 = 1 bit
    - 4 = 4 bits
    - 8 = 8 bits
    - 15 = 15 bits
    - 16 = 16 bits
    - 24 = 24 bits
    - 32 = 32 bits
    - 48 = 48 bits
  - **Message Inclusion:** Required
- **Field Name:** `browserScreenHeight`
  - **Description:** Total height of the cardholder’s screen in pixels. Value is returned from the *screen.height* property
  - **Length:** Variable, 1-6 characters
  - **Format:** String
  - **Accepted Value:**
  - **Message Inclusion:** Required

- **Field Name:** `browserScreenWidth`
  - **Description:** Total width of the cardholder's screen in pixels. Value is returned from the *screen.width* property
  - **Length:** Variable, 1-6 characters
  - **Format:** String
  - **Accepted Value:**
  - **Message Inclusion:** Required
- **Field Name:** `browserTZ`
  - **Description:** Time difference between UTC time and the cardholder's browser local time, in minutes
  - **Length:** Variable, 1-5 characters
  - **Format:** String
  - **Accepted Value:** Value is returned from the getTimezoneOffset() method
  - **Message Inclusion:** Required
- **Field Name:** `browserUserAgent`
  - **Description:** Exact content of the HTTP user-agent header
  - **Length:** Variable, maximum 2048 characters
  - **Format:** String
  - **Accepted Value:** If the total length of the User-Agent sent by the browser exceeds 2048 characters, the 3DS Server truncates the excess portion
  - **Message Inclusion:** Required
- **Field Name:** `cardExpiryDate`
  - **Description:** Expiry date of the PAN or token supplied to the 3DS Requestor by the cardholder
  - **Length:** 4 characters
  - **Format:** String (accepted format: YYMM)
  - **Accepted Value:**
  - **Message Inclusion:** Conditional
  - **Conditional Inclusion:** The requirements for the presence of this field are DS specific

- **Field Name:** `acctInfo`
  - **Description:** Additional information about the cardholder's account provided by the 3DS Requestor, described in acctInfo Data Elements
  - **Length:** Variable
  - **Format:** Object
  - **Accepted Value:**
  - **Message Inclusion:** Optional
- **Field Name:** `acctNumber`
  - **Description:** Account number that will be used in the authorisation request for payment transactions. May be represented by PAN, token.
  - **Length:** Variable, 13-19 characters, or up to 72 characters when encryption for sensitive data is enabled
  - **Format:** String
  - **Accepted Value:** Format represented in ISO 7812.
  - **Message Inclusion:** Required
- **Field Name:** `acctID`
  - **Description:** Additional information about the account optionally provided by the 3DS Requestor
  - **Length:** Variable, maximum 64 characters
  - **Format:** String
  - **Accepted Value:**
  - **Message Inclusion:** Optional
- **Field Name:** `billAddrCity`
  - **Description:** The city of the cardholder billing address associated with the card used for this transaction
  - **Length:** Variable, maximum 50 characters
  - **Format:** String
  - **Accepted Value:**
  - **Message Inclusion:** Conditional

- **Conditional Inclusion:** 01-PA: Required unless market or regional mandate restricts sending this information, 02-NPA: Required (if available) unless market or regional mandate restricts sending this information.

- **Field Name:** `billAddrCountry`
  - **Description:** The country of the cardholder billing address associated with the card used for this transaction
  - **Length:** 3 characters
  - **Format:** String
  - **Accepted Value:** ISO 3166-1 numeric three-digit country code, other than exceptions listed in Table A.5.
  - **Message Inclusion:** Conditional
  - **Conditional Inclusion:** Required if cardholder billing address state is present. 01-PA: Required unless market or regional mandate restricts sending this information. 02-NPA: Required (if available) unless market or regional mandate restricts sending this information.
- **Field Name:** `billAddrLine1`
  - **Description:** First line of the street address or equivalent local portion of the cardholder billing address associated with the card used for this transaction
  - **Length:** Variable, maximum 50 characters
  - **Format:** String
  - **Accepted Value:**
  - **Message Inclusion:** Conditional
  - **Conditional Inclusion:** 01-PA: Required unless market or regional mandate restricts sending this information, 02-NPA: Required (if available) unless market or regional mandate restricts sending this information.
- **Field Name:** `billAddrLine2`
  - **Description:** Second line of the street address or equivalent local portion of the cardholder billing address associated with the card used for this transaction
  - **Length:** Variable, maximum 50 characters
  - **Format:** String
  - **Accepted Value:**

- **Message Inclusion:** Conditional
- **Conditional Inclusion:** 01-PA: Required unless market or regional mandate restricts sending this information, 02-NPA: Required (if available) unless market or regional mandate restricts sending this information.

- **Field Name:** `billAddrLine3`
  - **Description:** Third line of the street address or equivalent local portion of the cardholder billing address associated with the card used for this transaction
  - **Length:** Variable, maximum 50 characters
  - **Format:** String
  - **Accepted Value:**
  - **Message Inclusion:** Conditional
  - **Conditional Inclusion:** 01-PA: Required unless market or regional mandate restricts sending this information, 02-NPA: Required (if available) unless market or regional mandate restricts sending this information.
- **Field Name:** `billAddrPostCode`
  - **Description:** ZIP or other postal code of the cardholder billing address associated with the card used for this transaction
  - **Length:** Variable, maximum 16 characters
  - **Format:** String
  - **Accepted Value:**
  - **Message Inclusion:** Conditional
  - **Conditional Inclusion:** 01-PA: Required unless market or regional mandate restricts sending this information, 02-NPA: Required (if available) unless market or regional mandate restricts sending this information.
- **Field Name:** `billAddrState`
  - **Description:** The state or province of the cardholder billing address associated with the card used for this transaction
  - **Length:** Variable, maximum 16 characters
  - **Format:** String
  - **Accepted Value:** Country subdivision code defined in ISO 3166-2
  - **Message Inclusion:** Conditional

- **Conditional Inclusion:** 01-PA: Required unless market or regional mandate restricts sending this information, or state is not applicable for this country. 02-NPA: Required (if available) unless market or regional mandate restricts sending this information, or State is not applicable for this country.

- **Field Name:** `email`
  - **Description:** The email address associated with the account that is either entered by the cardholder or is on file with the 3DS Requestor
  - **Length:** Variable, maximum 254 characters
  - **Format:** String
  - **Accepted Value:** To meet the requirements of Section 3.4 of IETF RFC 5322.
  - **Message Inclusion:** Conditional
  - **Conditional Inclusion:** Required unless market or regional mandate restricts sending this information
- **Field Name:** `homePhone`
  - **Description:** The home phone number provided by the cardholder, described in homePhone Data Elements
  - **Message Inclusion:** Conditional
  - **Conditional Inclusion:** Required unless market or regional mandate restricts sending this information
- **Field Name:** `mobilePhone`
  - **Description:** The mobile phone number provided by the cardholder, described in mobilePhone Data Elements
  - **Message Inclusion:** Conditional
  - **Conditional Inclusion:** Required unless market or regional mandate restricts sending this information
- **Field Name:** `cardholderName`
  - **Description:** Name of the cardholder
  - **Length:** Variable, 2-45 characters, or up to 72 characters when encryption for sensitive data is enabled
  - **Format:** String
  - **Accepted Value:** Alphanumeric special characters, listed in EMV Book 4, “Appendix B”.

- **Message Inclusion:** Conditional
- **Conditional Inclusion:** Required unless market or regional mandate restricts sending this information

- **Field Name:** `shipAddrCity`
  - **Description:** City portion of the shipping address requested by the cardholder
  - **Length:** Variable, maximum 50 characters
  - **Format:** String
  - **Accepted Value:**
  - **Message Inclusion:** Conditional
  - **Conditional Inclusion:** Required unless market or regional mandate restricts sending this information
- **Field Name:** `shipAddrCountry`
  - **Description:** Country of the shipping address requested by the cardholder
  - **Length:** 3 characters
  - **Format:** String
  - **Accepted Value:** ISO 3166-1 three-digit country code
  - **Message Inclusion:** Conditional
  - **Conditional Inclusion:** Required if cardholder shipping address state is present. Required, if available, unless market or regional mandate restricts sending this information.
- **Field Name:** `shipAddrLine1`
  - **Description:** First line of the street address or equivalent local portion of the shipping address requested by the cardholder
  - **Length:** Variable, maximum 50 characters
  - **Format:** String
  - **Accepted Value:**
  - **Message Inclusion:** Conditional
  - **Conditional Inclusion:** Required unless market or regional mandate restricts sending this information

- **Field Name:** `shipAddrLine2`
  - **Description:** Second line of the street address or equivalent local portion of the shipping address requested by the cardholder
  - **Length:** Variable, maximum 50 characters
  - **Format:** String
  - **Accepted Value:**
  - **Message Inclusion:** Conditional
  - **Conditional Inclusion:** Required unless market or regional mandate restricts sending this information
- **Field Name:** `shipAddrLine3`
  - **Description:** Third line of the street address or equivalent local portion of the shipping address requested by the cardholder
  - **Length:** Variable, maximum 50 characters
  - **Format:** String
  - **Accepted Value:**
  - **Message Inclusion:** Conditional
  - **Conditional Inclusion:** Required unless market or regional mandate restricts sending this information
- **Field Name:** `shipAddrPostCode`
  - **Description:** The ZIP or other postal code of the shipping address requested by the cardholder
  - **Length:** Variable, maximum 16 characters
  - **Format:** String
  - **Accepted Value:**
  - **Message Inclusion:** Conditional
  - **Conditional Inclusion:** Required unless market or regional mandate restricts sending this information
- **Field Name:** `shipAddrState`
  - **Description:** The state or province of the shipping address associated with the card being used for this transaction

- **Length:** Variable, maximum 3 characters
- **Format:** String
- **Accepted Value:** Country subdivision code defined in ISO 3166-2
- **Message Inclusion:** Conditional
- **Conditional Inclusion:** Required, if available, unless market or regional mandate restricts sending this information or state is not applicable for this country.

- **Field Name:** `workPhone`
  - **Description:** The work phone number provided by the cardholder. Described in workPhone Data Elements
  - **Message Inclusion:** Conditional
  - **Conditional Inclusion:** Required unless market or regional mandate restricts sending this information
- **Field Name:** `deviceChannel`
  - **Description:** Indicates the type of channel interface being used to initiate the transaction
  - **Length:** 2 characters
  - **Format:** String
  - **Accepted Value:**
    - 01 = App-based (APP)
    - 02 = Browser (BRW)
    - 03 = 3DS Requestor Initiated (3RI)
    - 04–79 = Reserved for EMVCo future use (values invalid until defined by EMVCo)
    - 80–99 = Reserved for DS use
  - **Message Inclusion:** Required
- **Field Name:** `deviceInfo`
  - **Description:** Device information gathered by the 3DS SDK from a consumer device as Base64url encoded JSON name/value pairs. This will be obtained from the SDK as encrypted data and populated by the DS, as unencrypted data, to the ACS.
  - **Length:** Variable, maximum 64000 characters
  - **Format:** Object
  - **Accepted Value:** Base64url encoded JSON object

- **Message Inclusion:** Conditional
- **Conditional Inclusion:** Required between the DS and ACS but will not be present from 3DS Server to DS

- **Field Name:** `deviceRenderOptions`
  - **Description:** Defines the SDK UI types that the device supports for displaying specific challenge user interfaces within the SDK, described in deviceRenderOptions Data Elements
  - **Length:** Variable, maximum 64000 characters
  - **Format:** JSON object
  - **Message Inclusion:** Required
- **Field Name:** `dsReferenceNumber`
  - **Description:** EMVCo assigned unique identifier to track approved DS
  - **Length:** Variable, maximum 32 characters
  - **Format:** String
  - **Accepted Value:**
  - **Message Inclusion:** Conditional
  - **Conditional Inclusion:** The DS will populate the AReq with this data element prior to passing it to the ACS
- **Field Name:** `dsTransID`
  - **Description:** Universally unique transaction identifier assigned by the DS to identify a single transaction
  - **Length:** 36 characters
  - **Format:** String
  - **Accepted Value:** Canonical format as defined in IETF RFC 4122. May utilise any of the specified versions as long as the output meets specified requirements
  - **Message Inclusion:** Conditional
  - **Conditional Inclusion:** The DS will populate the AReq with this data element prior to passing to the ACS. Required in error messages if available (e.g. can be obtained from a message or is generated).

- **Field Name:** `dsURL`
  - **Description:** URL of the DS to which the ACS will send the RReq if a challenge occurs. The ACS is responsible for storing this value for later use in the transaction for sending the RReq to the DS.
  - **Length:** Variable, maximum 2048 characters
  - **Format:** String
  - **Accepted Value:**
  - **Message Inclusion:** Conditional
  - **Conditional Inclusion:** Required between the DS and ACS but will not be present between the 3DS Server and DS
- **Field Name:** `payTokenInd`
  - **Description:** A value of `true` indicates that the transaction was detokenised prior to being received by the ACS. This data element will be populated by the system residing in the 3-D Secure domain where the detokenisation occurs (i.e. the 3DS Server or the DS). Note: The Boolean value of `true` is the only valid response for this field when it is present.
  - **Length:**
  - **Format:** Boolean
  - **Accepted Value:** true
  - **Message Inclusion:** Conditional
  - **Conditional Inclusion:** Required if there is detokenisation of an account number
- **Field Name:** `purchaseInstalData`
  - **Description:** Indicates the maximum number of authorisations permitted for instalment payments
  - **Length:** Variable, maximum 3 characters
  - **Format:** String
  - **Accepted Value:** Value greater than 1
  - **Message Inclusion:** Conditional
  - **Conditional Inclusion:** Required if the merchant and cardholder have agreed to instalment payments, i.e. if 3DS Requestor Authentication Indicator = 03. Omitted if not an instalment payment authentication.

- **Field Name:** `mcc`
  - **Description:** DS specific code describing the merchant's type of business, product or service
  - **Length:** 4 characters
  - **Format:** String
  - **Accepted Value:** This value correlates to the merchant category code as defined by each payment system or DS
  - **Message Inclusion:** 1-PA: Required, 02-NPA: Optional
- **Field Name:** `merchantCountryCode`
  - **Description:** Country code of the merchant. This value correlates to the merchant country code as defined by each payment system or DS
  - **Length:** 3 characters
  - **Format:** String
  - **Accepted Value:** ISO 3166-1 numeric three-digit country code. The same value must be used in the authorisation request.
  - **Message Inclusion:** 1-PA: Required, 02-NPA: Optional
- **Field Name:** `merchantName`
  - **Description:** Merchant name assigned by the acquirer or payment system
  - **Length:** Variable, maximum 40 characters
  - **Format:** String
  - **Accepted Value:** Same name used in the authorisation message as defined in ISO 8583
  - **Message Inclusion:** 1-PA: Required, 02-NPA: Optional
- **Field Name:** `merchantRiskIndicator`
  - **Description:** Merchant's assessment of the level of fraud risk for the specific authentication for both the cardholder and the authentication being conducted, described in merchantRiskIndicator Data Elements
  - **Length:** Variable
  - **Format:** Object
  - **Accepted Value:** Data will be formatted into a JSON object prior to being placed into the device merchant risk indicator field of the message.

- **Message Inclusion:** Optional

- **Field Name:** `messageCategory`
  - **Description:** Identifies the category of the message for a specific use case
  - **Length:** 2 characters
  - **Format:** String
  - **Accepted Value:**
    - 01-PA
    - 02-NPA
  - **Message Inclusion:** Required
- **Field Name:** `messageExtension`
  - **Description:** Data necessary to support requirements not otherwise defined in the 3-D Secure message are carried in a message extension, described in messageExtension Data Elements
  - **Length:** Variable, maximum 81920 bytes
  - **Format:** Array
  - **Accepted Value:**
  - **Message Inclusion:** Conditional
  - **Conditional Inclusion:** Conditions to be set by each DS.
- **Field Name:** `messageVersion`
  - **Description:** Protocol version identifier. This is version of the protocol specification utilised by the system creating this message. The message version number is set by the protocol version of the 3DS Server from which AReq message originates. The message version number does not change during a 3DS transaction.
  - **Length:** Variable, 5-8 characters
  - **Format:** String
  - **Accepted Value:**
    - 2.0.0 (deprecated)
    - 2.1.0 (active)
  - **Message Inclusion:** Required

- **Field Name:** `notificationURL`
  - **Description:** Fully qualified URL of the system that receives the CRes message or error message. The CRes message is posted by the ACS through the cardholder browser at the end of the challenge and on receipt of the RRes message.
  - **Length:** Variable, maximum 256 characters
  - **Format:** String
  - **Accepted Value:** Fully qualified URL
  - **Message Inclusion:** Required
- **Field Name:** `purchaseAmount`
  - **Description:** Purchase amount in minor units of currency with all punctuation removed. When used in conjunction with the purchase currency exponent field, proper punctuation can be calculated.
  - **Length:** Variable, 48 characters
  - **Format:** String
  - **Accepted Value:** Example: If the purchase amount is USD 123.45, element will contain the value 12345
  - **Message Inclusion:** 1-PA: Required, 02-NPA: Conditional
  - **Conditional Inclusion:** Required for 02-NPA if 3DS Requestor Authentication Indicator = 02 or 03
- **Field Name:** `purchaseCurrency`
  - **Description:** Currency in which the purchase amount is expressed.
  - **Length:** 3 characters
  - **Format:** String
  - **Accepted Value:** ISO 4217 three-digit currency code
  - **Message Inclusion:** 1-PA: Required, 02-NPA: Conditional
  - **Conditional Inclusion:** Required for 02-NPA if 3DS Requestor Authentication Indicator = 02 or 03
- **Field Name:** `purchaseExponent`
  - **Description:** Minor units of currency as specified in the ISO 4217 currency exponent
  - **Length:** 1 character

- **Format:** String
- **Accepted Value:**
- **Message Inclusion:** 1-PA: Required, 02-NPA: Conditional
- **Conditional Inclusion:** Required for 02-NPA if 3DS Requestor Authentication Indicator = 02 or 03

- **Field Name:** `purchaseDate`
  - **Description:** Date and time of the purchase, expressed in UTC.
  - **Length:** 14 characters
  - **Format:** String (Date Format: YYYYMMDDHHMMSS)
  - **Accepted Value:**
  - **Message Inclusion:** 1-PA: Required, 02-NPA: Conditional
  - **Conditional Inclusion:** Required for 02-NPA if 3DS Requestor Authentication Indicator = 02 or 03
- **Field Name:** `recurringExpiry`
  - **Description:** Date after which no further authorisations shall be performed
  - **Length:** 8 characters
  - **Format:** String (Date Format: YYYYMMDDHH)
  - **Accepted Value:**
  - **Message Inclusion:** 1-PA: Conditional, 02-NPA: Conditional
  - **Conditional Inclusion:** Required if 3DS Requestor Authentication Indicator = 02 or 03
- **Field Name:** `recurringFrequency`
  - **Description:** Indicates the minimum number of days between authorisations.
  - **Length:** Variable, 4 characters
  - **Format:** String
  - **Accepted Value:**
  - **Message Inclusion:** 1-PA: Conditional, 02-NPA: Conditional
  - **Conditional Inclusion:** Required if 3DS Requestor Authentication Indicator = 02 or 03

- **Field Name:** `sdkAppID`
  - **Description:** Universally unique ID created upon all installations and updates of the 3DS Requestor App on a Consumer Device. This will be newly generated and stored by the 3DS SDK for each installation or update
  - **Length:** 36 characters
  - **Format:** String
  - **Accepted Value:** Canonical format as defined in IETF RFC 4122. This may utilise any of the specified versions as long as the output meets specified requirements.
  - **Message Inclusion:** Required
- **Field Name:** `sdkEphemPubKey`
  - **Description:** Public key component of the ephemeral key pair generated by the 3DS SDK and used to establish session keys between the 3DS SDK and ACS
  - **Length:** Variable, 256 characters
  - **Format:** Object, JWK
  - **Accepted Value:**
  - **Message Inclusion:** Required
- **Field Name:** `sdkMaxTimeout`
  - **Description:** Indicates maximum amount of time, in minutes, for all exchanges.
  - **Length:** 2 characters
  - **Format:** String
  - **Accepted Value:** Greater than or = 5
  - **Message Inclusion:** Required
- **Field Name:** `sdkReferenceNumber`
  - **Description:** Identifies the vendor and version for the 3DS SDK that is integrated in a 3DS Requestor App, assigned by EMVCo when the 3DS SDK is approved
  - **Length:** Variable, 32 characters
  - **Format:** String
  - **Accepted Value:**
  - **Message Inclusion:** Required

- **Field Name:** `sdkTransID`
  - **Description:** Universally unique transaction identifier assigned by the 3DS SDK to identify a single transaction.
  - **Length:** 36 characters
  - **Format:** String
  - **Accepted Value:** Canonical format as defined in IETF RFC 4122. This may utilise any of the specified versions as long as the output meets specified requirements
  - **Message Inclusion:** Required
- **Field Name:** `transType`
  - **Description:** Identifies the type of transaction being authenticated.
  - **Length:** 2 characters
  - **Format:** String
  - **Accepted Value:**
    - 01 = Goods/ Service Purchase
    - 03 = Check Acceptance
    - 10 = Account Funding
    - 11 = Quasi-Cash Transaction
    - 28 = Prepaid Activation and Load. Note: Values derived from the 8583 ISO Standard.
  - **Message Inclusion:** Conditional
  - **Conditional Inclusion:** This field is required in some markets (e.g. for merchants in Brazil), otherwise, optional

---

threeDSRequestorAuthenticationInfo Data Elements

- **Field Name:** `threeDSReqAuthData`
  - **Description:** Data that documents and supports a specific authentication process. In the current version of the specification, this data element is not defined in detail, however the intention is that for each 3DS requestor authentication method, this field will carry the data for the ACS to use to verify the authentication process. For example, for method: 02—field can carry generic 3DS Requestor authentication information 03—data element can carry information about the provider of the federated ID and related

information 04—data element can carry the FIDO attestation data (including the signature) In future versions of the specification, these details are expected to be included

- **Length:** Variable, maximum 2048 bytes
- **Format:** String
- **Accepted Value:** Any
- **Message Inclusion:**

- **Field Name:** `threeDSReqAuthMethod`
  - **Description:** Mechanism used by the cardholder to authenticate to the 3DS Requestor
  - **Length:** 2 characters
  - **Format:** String
  - **Accepted Value:**
    - 01 = No 3DS Requestor authentication occurred (i.e. cardholder “logged in as guest”)
    - 02 = Login to the cardholder account at the 3DS Requestor system using 3DS Requestor’s own credentials
    - 03 = Login to the cardholder account at the 3DS Requestor system using a federated ID
    - 04 = Login to the cardholder account at the 3DS Requestor system using issuer credentials
    - 05 = Login to the cardholder account at the 3DS Requestor system using third-party authentication
    - 06 = Login to the cardholder account at the 3DS Requestor system using a FIDO Authenticator
    - 07–79 = Reserved for EMVCo future use (values invalid until defined by EMVCo)
    - 80–99 = Reserved for DS use
- **Field Name:** `threeDSReqAuthTimestamp`
  - **Description:** Date and time in UTC of the cardholder authentication
  - **Length:** 12 characters
  - **Format:** String (Date Format: YYYYMMDDHHMM)
  - **Accepted Value:**

threeDSRequestorPriorAuthenticationInfo Data Elements

- **Field Name:** `threeDSReqPriorAuthData`
  - **Description:** Data that documents and supports a specific authentication process. In the current version of the specification this data element is not defined in detail, however the intention is that for each 3DS requestor authentication method, this field carry data that the ACS can use to verify the authentication process. In future versions of the specification, these details are expected to be included.
  - **Length:** Maximum 2048 bytes
  - **Format:** Any
  - **Accepted Value:**
- **Field Name:** `threeDSReqPriorAuthMethod`
  - **Description:** Mechanism used by the cardholder to previously authenticate to the 3DS Requestor
  - **Length:** 2 characters
  - **Format:** String
  - **Accepted Value:**
    - 01 = Frictionless authentication by ACS
    - 02 = Cardholder challenge by ACS
    - 03 = AVS verified 04 = Other issuer methods
    - 05–79 = Reserved for EMVCo future use (values invalid until defined by EMVCo)
    - 80–99 = Reserved for DS use
- **Field Name:** `threeDSReqPriorAuthTimestamp`
  - **Description:** Date and time in UTC of the prior cardholder authentication
  - **Length:** 12 characters
  - **Format:** String (Date Format: YYYYMMDDHHMM)
  - **Accepted Value:**
- **Field Name:** `threeDSReqPriorRef`
  - **Description:** This data element provides additional information to the ACS to determine the best approach for handling a request

- **Length:** 36 characters
- **Format:** String
- **Accepted Value:** This data element contains an ACS transaction ID for a prior authenticated transaction (for example, the first recurring transaction that was authenticated with the cardholder)

acctInfo Data Elements

- **Field Name:** `chAccAgeInd`
  - **Description:** Length of time that the cardholder has had the account with the 3DS Requestor
  - **Length:** 2 characters
  - **Format:** String
  - **Accepted Value:**
    - 01 = No account (guest check-out)
    - 02 = Created during this transaction
    - 03 = Less than 30 days
    - 04 = 30–60 days
    - 05 = More than 60 days
- **Field Name:** `chAccChange`
  - **Description:** Date that the cardholder's account information with the 3DS Requestor was last changed. Information includes billing or shipping address, new payment account, or new user/s added
  - **Length:** 8 characters
  - **Format:** String (Date Format: YYYYMMDD)
  - **Accepted Value:**
- **Field Name:** `chAccChangeInd`
  - **Description:** Length of time since the cardholder's account information with the 3DS Requestor was last changed. Information includes billing or shipping address, new payment account, or new user/s added.
  - **Length:** 2 characters

- **Format:** String
- **Accepted Value:**
  - 01 = Changed during this transaction
  - 02 = Less than 30 days
  - 03 = 30–60 days
  - 04 = More than 60 days

- **Field Name:** `chAccDate`
  - **Description:** Date that the cardholder opened the account with the 3DS Requestor
  - **Length:** 8 characters
  - **Format:** String (Date Format: YYYYMMDD)
  - **Accepted Value:**
- **Field Name:** `chAccPwChange`
  - **Description:** Date that cardholder's account with the 3DS Requestor had a password change or account reset
  - **Length:** 8 characters
  - **Format:** String (Date Format: YYYYMMDD)
  - **Accepted Value:**
- **Field Name:** `chAccPwChangeInd`
  - **Description:** Indicates the length of time since the cardholder's account with the 3DS Requestor had a password change or account reset
  - **Length:** 2 characters
  - **Format:** String
  - **Accepted Value:**
    - 01 = No change
    - 02 = Changed during this transaction
    - 03 = Less than 30 days
    - 04 = 30–60 days
    - 05 = More than 60 days

- **Field Name:** `nbPurchaseAccount`
  - **Description:** Number of purchases using this cardholder account during the last six months
  - **Length:** Maximum 4 characters
  - **Format:** String
  - **Accepted Value:**
- **Field Name:** `provisionAttemptsDay`
  - **Description:** Number of attempts made to add a card in the last 24 hours
  - **Length:** Maximum 3 characters
  - **Format:** String
  - **Accepted Value:**
- **Field Name:** `txnActivityDay`
  - **Description:** Number of transactions (successful and abandoned), in the last 24 hours, for this cardholder's account with the 3DS Requestor, across all payment accounts
  - **Length:** Maximum 3 characters
  - **Format:** String
  - **Accepted Value:**
- **Field Name:** `txnActivityYear`
  - **Description:** Number of transactions (successful and abandoned), in the last year, for this cardholder account with the 3DS Requestor, across all payment accounts
  - **Length:** Maximum 3 characters
  - **Format:** String
  - **Accepted Value:**
- **Field Name:** `paymentAccAge`
  - **Description:** Date that the payment account was enrolled in the cardholder’s account with the 3DS Requestor
  - **Length:** 8 characters
  - **Format:** String (Date Format: YYYYMMDD)
  - **Accepted Value:**

- **Field Name:** `paymentAccInd`
  - **Description:** Indicates the length of time that the payment account was enrolled in the cardholder's account with the 3DS Requestor
  - **Length:** 2 characters
  - **Format:** String
  - **Accepted Value:**
    - 01 = No account (guest check-out)
    - 02 = During this transaction
    - 03 = Less than 30 days
    - 04 = 30–60 days
    - 05 = More than 60 days
- **Field Name:** `shipAddressUsage`
  - **Description:** Date when the shipping address used for this transaction was first used with the 3DS Requestor
  - **Length:** 8 characters
  - **Format:** String (Date format = YYYYMMDD)
  - **Accepted Value:**
- **Field Name:** `shipAddressUsageInd`
  - **Description:** Indicates when the shipping address used for this transaction was first used with the 3DS Requestor
  - **Length:** 2 characters
  - **Format:** String
  - **Accepted Value:**
    - 01 = This transaction
    - 02 = Less than 30 days
    - 03 = 30–60 days
    - 04 = More than 60 days
- **Field Name:** `shipNameIndicator`
  - **Description:** Indicates if the cardholder name on the account is identical to the shipping name used for this transaction

- **Length:** 2 characters
- **Format:** String
- **Accepted Value:**
  - 01 = Account name identical to shipping name
  - 02 = Account name different to shipping name

- **Field Name:** `suspiciousAccActivity`
  - **Description:** Indicates whether the 3DS Requestor has experienced suspicious activity (including previous fraud) on the cardholder account
  - **Length:** 2 characters
  - **Format:** String
  - **Accepted Value:**
    - 01 = No suspicious activity has been observed
    - 02 = Suspicious activity has been observed

homePhone Data Elements

- **Field Name:** `cc`
  - **Description:** Country code section of the number
  - **Length:** 1-3 characters
  - **Format:** String
  - **Accepted Value:** Refer to ITU-E.164 for additional information on format and length
  - **Message Inclusion:** Conditional
  - **Conditional Inclusion:** Required, if available, unless market or regional mandate restricts sending this information
- **Field Name:** `subscriber`
  - **Description:** Subscriber section of the number
  - **Length:** Variable, maximum 15 characters
  - **Format:** String
  - **Accepted Value:** Refer to ITU-E.164 for additional information on format and length
  - **Message Inclusion:** Conditional

- **Conditional Inclusion:** Required, if available, unless market or regional mandate restricts sending this information

mobilePhone Data Elements

- **Field Name:** `cc`
  - **Description:** Country code section of the number
  - **Length:** 1-3 characters
  - **Format:** String
  - **Accepted Value:** Refer to ITU-E.164 for additional information on format and length
  - **Message Inclusion:** Conditional
  - **Conditional Inclusion:** Required, if available, unless market or regional mandate restricts sending this information
- **Field Name:** `subscriber`
  - **Description:** Subscriber section of the number
  - **Length:** Variable, maximum 15 characters
  - **Format:** String
  - **Accepted Value:** Refer to ITU-E.164 for additional information on format and length
  - **Message Inclusion:** Conditional
  - **Conditional Inclusion:** Required, if available, unless market or regional mandate restricts sending this information

workPhone Data Elements

- **Field Name:** `cc`
  - **Description:** Country code section of the number
  - **Length:** 1-3 characters
  - **Format:** String
  - **Accepted Value:** Refer to ITU-E.164 for additional information on format and length
  - **Message Inclusion:** Conditional
  - **Conditional Inclusion:** Required, if available unless market or regional mandate restricts sending this information

- **Field Name:** `subscriber`
  - **Description:** Subscriber section of the number
  - **Length:** Variable, maximum 15 characters
  - **Format:** String
  - **Accepted Value:** Refer to ITU-E.164 for additional information on format and length
  - **Message Inclusion:** Conditional
  - **Conditional Inclusion:** Required, if available, unless market or regional mandate restricts sending this information

deviceRenderOptions Data Elements

- **Field Name:** `sdkInterface`
  - **Description:** Lists all SDK interface types supported by the device for displaying specific challenge user interfaces within the SDK
  - **Length:** 2 characters
  - **Format:** String
  - **Accepted Value:**
    - 01 = Native
    - 02 = HTML
    - 03 = Both
- **Field Name:** `sdkUiType`
  - **Description:** Lists all UI types supported by the device for displaying specific challenge user interfaces within the SDK
  - **Length:** 2 characters
  - **Format:** Array of String
  - **Accepted Value:**
    - 01 = Text
    - 02 = Single Select
    - 03 = Multi Select
    - 04 = OOB

        - 05 = HTML Other (valid only for HTML UI)

merchantRiskIndicator Data Elements

- **Field Name:** `deliveryEmailAddress`
  - **Description:** For electronic delivery, the email address to which the merchandise was delivered
  - **Length:** Maximum 254 characters
  - **Format:** String
  - **Accepted Value:**
- **Field Name:** `deliveryTimeframe`
  - **Description:** Indicates the merchandise delivery timeframe
  - **Length:** 2 characters
  - **Format:** String
  - **Accepted Value:**
    - 01 = Electronic delivery
    - 02 = Same day shipping
    - 03 = Overnight shipping
    - 04 = Two or more days shipping
- **Field Name:** `giftCardAmount`
  - **Description:** For prepaid or gift card purchases, the purchase amount total of prepaid or gift cards, in major units (for example, USD 123.45 is 123).
  - **Length:** Maximum 15 characters
  - **Format:** String
  - **Accepted Value:**
- **Field Name:** `giftCardCount`
  - **Description:** For prepaid or gift card purchases, the total count of individual prepaid or gift cards/codes purchased
  - **Length:** 2 characters
  - **Format:** String

- **Accepted Value:**

- **Field Name:** `giftCardCurr`
  - **Description:** For prepaid or gift card purchases, the currency code of the cards as defined in ISO 4217, other than those listed in Table A.5
  - **Length:** 3 characters
  - **Format:** String
  - **Accepted Value:**
- **Field Name:** `preOrderDate`
  - **Description:** For a pre-ordered purchase, the expected date that the merchandise will be available
  - **Length:** 8 characters
  - **Format:** String (Date format = YYYYMMDD)
  - **Accepted Value:**
- **Field Name:** `preOrderPurchaseInd`
  - **Description:** Indicates whether the cardholder is placing an order for merchandise with a future availability or release date
  - **Length:** 2 characters
  - **Format:** String
  - **Accepted Value:**
    - 01 = Merchandise available
    - 02 = Future availability
- **Field Name:** `reorderItemsInd`
  - **Description:** Indicates whether the cardholder is reordering previously purchased merchandise
  - **Length:** 2 characters
  - **Format:** String
  - **Accepted Value:**
    - 01 = First time ordered
    - 02 = Reordered

- **Field Name:** `shipIndicator`
  - **Description:** Indicates the shipping method chosen for the transaction. Merchants must choose the shipping indicator code that most accurately describes the cardholder's specific transaction, not their general business. If one or more items are included in the sale, use the shipping indicator code for the physical goods or if all goods are digital, use the shipping indicator code that describes the most expensive item.
  - **Length:** 2 characters
  - **Format:** String
  - **Accepted Value:**
    - 01 = Ship to cardholder’s billing address
    - 02 = Ship to another verified address on file with merchant
    - 03 = Ship to address that is different to the cardholder’s billing address
    - 04 = “Ship to store” / pick-up at local store (Store address to be populated in shipping address fields)
    - 05 = Digital goods (includes online services, electronic gift cards and redemption codes)
    - 06 = Travel and event tickets, not shipped
    - 07 = Other (for example, gaming, digital services not shipped, media subscriptions, etc.)

messageExtension Data Elements

- **Field Name:** `criticalityIndicator`
  - **Description:** A Boolean value indicating whether the recipient must understand the contents of the extension to interpret the entire message
  - **Length:**
  - **Format:** Boolean
  - **Accepted Value:** true, false
- **Field Name:** `data`
  - **Description:** The data carried in the extension
  - **Length:** Variable, maximum 8059 characters
  - **Format:** String (JSON text)

    - **Accepted Value:**

- **Field Name:** `id`
    - **Description:** A unique identifier for the extension. Note: the payment system registered application provider identifier (RID) is required as a prefix of the ID
    - **Length:** Variable, maximum 64 characters
    - **Format:** String
    - **Accepted Value:**
- **Field Name:** `name`
    - **Description:** The name of the extension data set as defined by the extension owner
    - **Length:** Variable, maximum 64 characters
    - **Format:** String
    - **Accepted Value:**

AdditionalInfo Data Elements

- **Field Name:** `clientId`
    - **Description:** Client ID
    - **Length:** 15 characters
    - **Format:** String
    - **Accepted Value:** Decimal numbers
    - **Message Inclusion:** Optional (this field will be excluded from RESTful JSON message when it has no value)

Whitelisting

New page added.

The Whitelisting feature in ActiveAccess enables cardholders to add 3DS Requestors/merchants to a trusted beneficiaries list. The exposed whitelisting APIs are used to make changes to an issuer's cardholders' whitelisted merchants. Additionally, the APIs enable retrieving data by searching through the whitelists, and provide the ability to access a history of changes applied for cardholders.

Message Format

All Whitelisting APIs accept *JSON* requests and responses. The supported protocol to exchange data is *HTTP* or *HTTPS*, depending on the configurations of your application server (e.g Tomcat).

API Description

Add Cardholder

This API is used to add a new cardholder to the whitelist. A card number can be added to the whitelist of a merchant which is defined for an issuer.

- **Path:** whitelisting/wl/api/merchant/add
- **Method Type:** POST
- **Request Body (sample):**

```
{
   "merchantName": "Test Merchant",
   "mcc": "3200",
   "merchantCountryCode": "840",
   "acquirerMerchantID": "123456789012345",
   "issuerName": "Any Bank",
   "cardNumber": "12345678905",
   "cardName": "cardholder name"
}
```

- **Field Description:**
  - **merchantName:** Name of the merchant. This field is required.
  - **mcc:** Merchant Category Code. This field is required.
  - **merchantCountryCode:** Country code of the merchant. This field is required.
  - **acquirerMerchantID:** Acquirer's Merchant ID. This field is required.
  - **issuerName:** Name of the issuer. The issuer must exist on ActiveAccess. Instead of issuerName, you can also pass **issuerId** which is the Issuer ID of an issuer on ActiveAccess. This field is required.
  - **cardNumber:** Card number of the cardholder. This field is required.
  - **cardName:** Name of the cardholder. This field can be blank, i.e. `"cardName" : ""` . If you do not pass a cardholder name to the API or set a null value for it, it will be assumed that all the cardholder names of the card number should be considered as whitelisted. Additionally, you cannot add any other cards with that card number but different cardholder names to the merchant.

- **Successful Response:**

```
{"status":"SUCCESS"}
```

- **Error Response - required fields missing:**

```
{
"status": "ERROR",
"messageLabel": "MISSED_REQUIRED_FIELD",
"fields": "merchantName, mcc, merchantCountryCode, acquirerMerchantID,
cardNumber, issuerName"
}
```

- **Error Response - incorrect issuer name:**

```
{
 "status": "ERROR",
 "messageLabel": "NOT_FOUND",
 "fields": "issuerName"
}
```

- **Error Response - incorrect issuer ID:**

```
{
 "status": "ERROR",
 "messageLabel": "NOT_FOUND",
 "fields": "issuerId"
}
```

- **Error Response - adding an existing merchant:**

```
 {
    "status": "ERROR",
    "messageLabel": "MERCHANT_CARDHOLDER_ALREADY_EXIST"
 }
```

Remove Merchant

This API is used to remove a merchant from the whitelist.

> **Warning**
>
> When the specified merchant is deleted, it will be removed from the whitelisted merchants of all the cardholders.

- **Path:** whitelisting/wl/api/merchant/remove
- **Method Type:** POST
- **Request Body (sample):**

```
{
   "merchantName": "Test Merchant",
   "mcc": "3200",
   "merchantCountryCode": "840",
   "acquirerMerchantID": "123456789012345",
   "issuerName": "Any Bank"
}
```

- **Fields Description:**
  - **merchantName:** Name of the merchant. This field is required.
  - **mcc:** Merchant Category Code. This field is required.
  - **merchantCountryCode:** Country code of the merchant. This field is required.
  - **acquirerMerchantID:** Acquirer's Merchant ID. This field is required.

- **issuerName:** Name of the issuer. The issuer must exist on ActiveAccess.

- **Successful Response:**

```
{"status":"SUCCESS"}
```

- **Error Response - required fields missing:**

```
{
 "status": "ERROR",
 "messageLabel": "MISSED_REQUIRED_FIELD",
 "fields": "merchantName, mcc, merchantCountryCode, acquirerMerchantID,
issuerName"
}
```

- **Error Response - incorrect issuer name:**

```
{
 "status": "ERROR",
 "messageLabel": "NOT_FOUND",
 "fields": "issuerName"
}
```

Remove Merchant List

This API is used to remove a list of merchants from the whitelist.

> **Warning**
>
> When the specified merchant is deleted, it will be removed from the whitelisted merchants of all the cardholders.

- **Path:** whitelisting/wl/api/merchant/removelist
- **Method Type:** POST
- **Request Body (sample):**

```
[
   {
      "issuerName": "Any Bank",
      "merchantName": "Test Merchant1",
      "mcc": "123",
      "merchantCountryCode": "123",
      "acquirerMerchantID": "89012345"
```

```
    },
    {
       "issuerName": "Any Bank",
       "merchantName": "Test Merchant3",
       "merchantCountryCode": "125",
       "acquirerMerchantID": "8901234545",
       "mcc": "125"
    }
]
```


- **Fields Description:**
  - **merchantName:** Name of the merchant. This field is required.
  - **mcc:** Merchant Category Code. This field is required.
  - **merchantCountryCode:** Country code of the merchant. This field is required.
  - **acquirerMerchantID:** Acquirer's Merchant ID. This field is required.
  - **issuerName:** Name of the issuer. The issuer must exist on ActiveAccess.
- **Successful Response:**


```
{
 "status": "SUCCESS",
 "failedMerchants": []
}
```


- **Error Response - required fields missing:**


```
{
 "status": "ERROR",
 "failedMerchants": [
  {
   "resMessageDto": {
   "status": "ERROR",
   "messageLabel": "MISSED_REQUIRED_FIELD",
   "fields": "mcc"
   },
   "reqMerchant": {
      "merchantName": "Test Merchant3",
      "mcc": null,
      "merchantCountryCode": "125",
      "acquirerMerchantID": "8901234545",
      "cardName": null,
      "cardNumber": null,
      "issuerName": "Any Bank"
   }
  }
```

```
 ]
}
```

- **Error Response - incorrect issuer name:**

  In the sample response below, all merchants in the request had correct issuer names, except one. All other merchants are deleted successfully.

```
{
 "status": "ERROR",
 "failedMerchants": [
  {
   "resMessageDto": {
   "status": "ERROR",
   "messageLabel": "NOT_FOUND",
   "fields": "issuerName"
    },
   "reqMerchant": {
      "merchantName": "Test Merchant3",
      "mcc": "125",
      "merchantCountryCode": "125",
      "acquirerMerchantID": "89012345",
      "cardName": null,
      "cardNumber": null,
      "issuerName": "Any Bank2"
   }
  }
 ]
}
```

Search Cardholder

This API is used to find cardholders and merchants on the whitelist. There are filters for different fields to fetch the desired results.

- **Path:** whitelisting/wl/api/merchant/getMerchant
- **Method Type:** POST
- **Query Parameters:**
  - **first:** First row which is fetched from the database. It starts from 0.
  - **size:** Maximum number of rows which are fetched from the database.
- **Request Body (sample):**

```
{
   "issuerName": "Any Bank",
   "merchantName": "Test Merchant",
   "merchantCountryCode": "840",
   "mcc": "3200",
   "acquirerMerchantID": "123456789012345",
   "cardName": "cardholder name"
   "cardNumber": "123456789056789",
   "onlyNullCardName": false
}
```

- **Fields Description:**
  - **merchantName:** Name of the merchant.
  - **mcc:** Merchant Category Code.
  - **merchantCountryCode:** Country code of the merchant.
  - **acquirerMerchantID:** Acquirer's Merchant ID.
  - **issuerName:** Name of the issuer. The issuer must exist on ActiveAccess. This field is required if `cardName` and `cardNumber` are being used in the search.
  - **cardNumber:** Card number of the cardholder.
  - **cardName:** Name of the cardholder. By default, cardholders with a null cardholder name value are included in the results.
  - **onlyNullCardName:** This field is used to specify whether the results should include data where `cardName` is null. If `onlyNullCardName` is set to true, only cards with a null `cardName` are included in the results. If `onlyNullCardName` is set to false, cards with a null cardName are not included in the results.
- **Successful Response:**

```
{
 "status": "SUCCESS",
 "response": [
   {
     "merchantName": "Test Merchant1",
     "mcc": "123",
     "merchantCountryCode": "123",
     "acquirerMerchantID": "89012345",
     "cardName": "cardholder name",
     "cardNumber": "123456789056789",
     "issuerName": "Any Bank"
  },
  {
```

```
      "merchantName": "Test Merchant1",
      "mcc": "123",
      "merchantCountryCode": "123",
      "acquirerMerchantID": "89012345",
      "cardName": null,
      "cardNumber": "123456789056789",
      "issuerName": "Any Bank"
    }
]
}
```


- **Error Response - cardName or cardNumber are in the request, however issuerName does not exist:**


```
{
 "status": "ERROR",
 "messageLabel": "MISSED_ISSUER_FOR_CARD",
 "fields": "issuerName"
}
```


- **Error Response - cardName is null and onlyNullCardName does not exist**


```
{
 "status": "ERROR",
 "messageLabel": "MISSED_ISSUER_FOR_CARD",
 "fields": "onlyNullCardName"
}
```


Search Cardholder History

This API is used to find a history of insertions and deletions of cardholders and merchants on the whitelist. There are filters for different fields to fetch the desired results.

- **Path:** whitelisting/wl/api/merchant/getMerchantHistory
- **Method Type:** POST
- **Query Parameters:**
  - **first:** First row which is fetched from the database. It starts from 0.
  - **size:** Maximum number of rows which are fetched from the database.
- **Request Body (sample):**

```
{
   "issuerName": "Any Bank",
   "merchantName": "Test Merchant",
   "merchantCountryCode": "840",
   "mcc": "3200",
   "acquirerMerchantID": "123456789012345",
   "cardName": "cardholder name"
   "cardNumber": "123456789056789",
   "onlyNullCardName": false,
   "fromDate": 1619385552712,
   "toDate": 1619385601308
}
```


- **Fields Description:**
  - **merchantName:** Name of the merchant.
  - **mcc:** Merchant Category Code.
  - **merchantCountryCode:** Country code of the merchant.
  - **acquirerMerchantID:** Acquirer's Merchant ID.
  - **issuerName:** Name of the issuer. The issuer must exist on ActiveAccess. This field is required if `cardName` and `cardNumber` are being used in the search.
  - **cardNumber:** Card number of the cardholder.
  - **cardName:** Name of the cardholder. By default, cardholders with a null cardholder name value are included in the results.
  - **onlyNullCardName:** This field is used to specify whether the results should include data where `cardName` is null. If `onlyNullCardName` is set to true, only cards with a null `cardName` are included in the results. If `onlyNullCardName` is set to false, cards with a null cardName are not included in the results.
- **Successful Response:**


```
  {
     "status": "SUCCESS",
     "response": [
        {
           "merchantName": "Test Merchant",
           "mcc": "3200",
           "merchantCountryCode": "840",
           "acquirerMerchantID": "123456789012345",
           "cardName": "cardholder name"
           "cardNumber": "12345678905",
           "issuerName": "Any Bank",
```

```
            "auditOperation": "INSERTED",
            "actionTime": "2021-04-25T21:19:36.063+00:00"
        },
        {
            "merchantName": "Test Merchant",
            "mcc": "3200",
            "merchantCountryCode": "840",
            "acquirerMerchantID": "123456789012345",
            "cardName": "cardholder name"
            "cardNumber": "12345678905",
            "issuerName": "Any Bank",
            "auditOperation": "DELETED",
            "actionTime": "2021-04-25T21:20:01.308+00:00"
        }
    ]
}
```


The `auditOperation` property indicates whether the merchant has been inserted or deleted.

- **Error Response - cardName or cardNumber are in the request, however issuerName does not exist:**


```
{
   "status": "ERROR",
   "messageLabel": "MISSED_ISSUER_FOR_CARD",
   "fields": "issuerName"
}
```


- **Error Response - cardName is null and onlyNullCardName does not exist**


```
{
 "status": "ERROR",
 "messageLabel": "MISSED_ISSUER_FOR_CARD",
 "fields": "onlyNullCardName"
}
```

Error Codes

Server Error Codes

Server Error Codes			
Code	**Message**	**Details**	**Usage**
1	Root element invalid.	Exception message and its cause FourDSecure ThreeDSecure	Yes
2	Message element not a defined message.	Exception message and its cause VVRQ PPRQ Undefined CRReq	Yes
3	Required element missing.	PaReq TermUrl MD Id | VEReq.Extension.Id | PAReq.Extension id VEReq.version | version | PAReq.version Pan | VEReq.Pan PAReq.Merchant.name | name PAReq.Merchant.country | country PAReq.Merchant.url | url PAReq.Purchase.xid | xid PAReq.Purchase.date | date PAReq.Purchase.amount | amount PAReq.Purchase.purchAmount | purchAmount PAReq.Purchase.currency | currency PAReq.Purchase.exponent | exponent PAReq.CH.acctID | acctID PAReq.CH.expiry | expiry Message.Id | Id Message	Yes
4	Critical element not recognized.	Extension | VEReq.Extension | PAReq.Extension	Yes

Server Error Codes			
5	Format of one or more elements is invalid according to the specification.	Exception message and its cause version | VEReq.Version | PAReq.Version Pan | VEReq.Pan VEReq.Extension.Id | Extension.Id VEReq.Browser.deviceCategory | devicCategory Extension.Critical PAReq.Merchant.name | name | Merchant.name PAReq.Merchant.country | country | Merchant.country PAReq.Purchase.xid | xid | Purchase.xid PAReq.Purchase.date | date | Purchase.date PAReq.Purchase.amount | amount | Purchase.amount PAReq.Purchase.purchAmount | purchAmount | Purchase.purchAmount PAReq.Purchase.currency | currency | Purchase.currency PAReq.Purchase.exponent | exponent | Purchase.exponent PAReq.Purchase.desc | desc | Purchase.desc PAReq.Purchase.Recur.frequency | frequency | Recur.frequency PAReq.Purchase.Recur.endRecur | endRecur | Purchase.Recur.endRecur PAReq.Purchase.install | install | .Purchase.install PAReq.CH.acctID | acctId | CH.acctID PAReq.CH.expiry | expiry | CH.expiry Message.Id | Id Merchant Merchant.merID	Yes
6	Protocol version too old.	Protocol version too old. Protocol version is not supported by ProtectBuy.	Yes
98	Transient system failure.	Contact your vendor with this 'ACS Session ID': %sessionId%	Yes
99	Permanent system failure.	%s	No

Server Error Codes			
101	Message Received Invalid	One of the following: • Invalid Message Type • Invalid Message for the receiving component • Invalid Formatted Message	yes
102	Message Version Number Not Supported	1.0.2, 2.1.0	yes
103	Sent Messages Limit Exceeded	Exceeded maximum number of PReq messages sent to the DS	no
201	Required Data Element Missing	%s	yes
202	Critical Message Extension Not Recognised	%s	yes
203	Format of one or more Data Elements is Invalid according to the Specification	%s	yes
204	Duplicate Data Element	%s	yes
301	Transaction ID Not Recognised	%s	yes
302	Data Decryption Failure	Data could not be decrypted by the receiving system due to technical or other reason	yes
303	Access Denied, Invalid Endpoint	Access denied, invalid endpoint	no
304	ISO Code Invalid	%s	yes
305	Transaction data not valid	%s	yes
306	Merchant Category Code (MCC) Not Valid for Payment System	Merchant Category Code (MCC) not valid for Payment System.	no
307	Serial Number not Valid	Serial Number not valid	no

Server Error Codes			
402	Transaction Timed Out	Transaction timed-out.	yes
403	Transient System Failure	%s	yes
404	Permanent System Failure	%s	yes
405	System Connection Failure	System connection failure.	yes
1001	Invalid http request	Invalid HTTP request: PAHndler.run() Invalid HTTP request:	Yes
1002	Process timed out	Process timed out	Yes
1003	Invalid xml request	Invalid XML request process.	No
1004	Error in ThreeDS.service(): %s	Error in ThreeDS.service(): %s	No
1005	Permission denied	Permission denied	Yes
1006	An extension is not currently associated with this request	An extension is not currently associated with this request	Yes
1007	ACS failed to start successfully.	ACS failed to start successfully	Yes
1008	Error in inflating PAReq	Error in inflating PAReq ver 1.0.1	Yes
1009	Error in deflating PARes	Error in deflating PARes ver 1.0.1	No
1010	This session is invalid. Please try again.	This session is invalid. Please try again.	Yes
1011	Your session has now expired. Please try again.	Your session has expired. Please try again.	Yes
1012	Internal error: Unable to save session.	Internal error: Unable to save session.	No

Server Error Codes			
1013	Invalid authentication result in ThreeDS.service(): %s	Invalid authentication result in ThreeDS.service(): %s	No
1014	'%s' request length is too large	'HTTP' request length is too large 'XML' request length is too large	Yes
1015	Invalid cardholder name for PARes 10X in ThreeDS.service()	Invalid cardholder name for PARes 10X in ThreeDS.service()	No
1016	The process has been successfully completed. One or more required parameters were not specified.	The process has been successfully completed. One or more required parameters were not specified.	Yes
1017	Cannot find any authentication data.	Authentication data not found.	Yes
1018	Issuer's BIN does not support device authentication over 3-D Secure.	This issuer BIN range does not support device authentication for 3-D Secure.	No
1019	Issuer does not support any devices.	Issuer does not support any devices.	Yes
1020	Invalid request.	ACS records show the card type is MasterCard but the request was received as on Visa VE server. ACS records show the card type Visa but the request was received as on MasterCard VE server. …	Yes
1021	There is no assigned device.	There is no device assigned.	Yes
1022	Different card types.	Cards belong to different card schemes.	Yes
1023	Invalid character	There is an invalid character in parameter (%s)	No
1024	Invalid card in authentication process	Card is pre-registered and cannot be used in the authentication process.	Yes

Server Error Codes			
1025	Illegal process	Illegal process 'Authorization'	Yes
1026	Server is in reinitializing state	Server is in reinitializing state.	Yes
1027	Invalid authentication URL	'Url' is invalid	Yes
1028	Cannot find all the required parameters for PA processing	Cannot find all the required parameters for PA processing 'URI'.	Yes
1029	Page and process do not match	The 'page name' page cannot be displayed while in the duplicate cardholder process.	Yes
1030	Invalid parameter value		No
1031	Email Device Param not initialized		Yes

User Error Codes

User Error Codes			
Code	**Message**	**Details**	**Usage**
1	Root element invalid.	Device	Yes
2	Message element not a defined message.	Name of undefined element	Yes
3	Required element missing.	Name of missing element	Yes
4	Critical element not recognized.	Extension	Yes
5	Format of one or more elements is invalid according to the specification.	Name of invalid element	Yes

User Error Codes			
50	Issuer %s does not participate in device authentication.	%s	Yes
55	Transaction data not valid.	%s	Yes
56	Signature verification failed.	%S	Yes
70	Invalid request	%S	Yes
71	Session is invalid.	%S	Yes
72	Session is expired.	%S	Yes
98	Transient system failure	%S	Yes
99	Permanent system failure.	%S	Yes
1001	Invalid HTTP request	Invalid request	No
1002	Process timed out	Process timed out	No
1003	Invalid XML request	Invalid XML request	No
1004		%s does not exist or has an incorrect format	No
1005	Permission denied	Permission denied	No
1006	An extension is not currently associated with this request	An extension is not currently associated with this request	No
1007	Server has not started correctly	Server has not started correctly	No
1008		Error in serializing the %s XML Document	No
1009		Session '%s' has expired	No

User Error Codes			
1010	Invalid request length	'%s' request length is too large	No
1011		The process has been successfully completed. One or more required parameters were not specified	No
1012		Error in inflating UAReq ver 1.0.1	No
1013		Error in deflating UARes ver 1.0.1	No
2001	User not registered		No
2002	User is locked		Yes
2003	Action cancelled		Yes
2004	User is disabled		Yes
2005	Maximum number of transactions exceeded		Yes
2010	Device not registered		Yes
2011	Cannot find any active devices		Yes
2012	Device type is not supported. Type = %s		Yes
2013	Invalid device extension, %s		Yes
2014	Invalid token		Yes
2015	Invalid password		Yes
2016	One-way authentication is not supported for device type %s		Yes
2017	Maximum number of SMS resend request exceeded		Yes

User Error Codes			
2050	Issuer %s does not participate in device authentication		Yes
2051	License key does not allow for device authentication, %s		Yes
2052	Invalid password for issuer %s		Yes
2053	Device type %s is not supported for issuer %s		Yes
2054	The interface is disabled for issuer %s		Yes
2055	Device type %s is not supported by the device owner (issuer: %s)		Yes
2056	The process has been successfully completed. One or more required parameters were not specified.		Yes
2057	Duplicate UAReq not allowed		Yes

Account Error Messages

Account Error Messages		
Code	**Message**	**Usage**
101	Please re-enter the field(s) highlighted in red	Yes
102	Required field missing	Yes
103	Invalid number	No
104	Invalid password	Yes

Account Error Messages		
105	Invalid activation code	No
106	Data verification error	Yes
107	Field length exceeded	Yes
108	Invalid one time password	Yes
109	Invalid cardholder name	Yes
110	Invalid cardholder	No
111	Invalid password length	No
112	Passwords do not match	Yes
113	Invalid answer	Yes
114	Invalid username	Yes
115	Invalid full name	Yes
116	Invalid personal assurance message (PAM)	Yes
117	Invalid expiry date	Yes
118	Invalid card number	Yes
120	Invalid question	No
121	Invalid device type selected	Yes
122	Resynchronization failed	Yes
123	Invalid cardID	Yes
124	Password must be between [%1] to [%2] characters long	Yes

Account Error Messages		
125	Password must contain at least [?] number(s)	Yes
126	Password must contain at least [?] capital letter(s)	Yes
127	Unicode characters cannot be used	Yes
128	Invalid character	No
129	The parameter ([?]) is required	Yes
130	Invalid PriSec	Yes
131	The Personal Message must not contain your Verified by Visa password or Password Hint	Yes
132	The Password Hint must not contain your Verified by Visa password	Yes
133	The account should have ([?]) authentication data	Yes
134	Invalid Hint	Yes
135	Invalid Data Format	Yes
136	[%1] does not match the confirmation [%2]	Yes

Authentication Device Messages

Authentication Device Messages		
Code	**Message**	**Usage**
101	Please re-enter the field(s) highlighted in red	No
102	Required field missing	Yes
103	Invalid number	No

Authentication Device Messages		
104	Invalid password	No
105	Invalid Activation Code	No
106	Data verification error	No
107	Field length exceeded	Yes
108	Invalid one time password	Yes
301	Current Token:	Yes
302	Please enter the one time password from one of your existing devices here	Yes
303	Invalid one time password	Yes
304	Invalid serial number	Yes
305	Device is lost	Yes
306	Device is damaged	Yes
307	Device is already assigned	Yes
401	Current Token:	No
402	Please enter the one time password from one of your existing devices here	No
403	Invalid one time password	Yes
404	Invalid serial number	Yes
405	Device is lost	Yes
406	Device is damaged	Yes
407	Device is already assigned	Yes

Authentication Device Messages		
501	SMS Token:	Yes
502	Please enter the one time password which was sent to you via SMS	Yes
503	Invalid SMS one time password	Yes
504	Invalid mobile number	Yes
505	Invalid mobile network provider	Yes
506	Invalid country calling code	Yes
507	Please enter the mobile number only, without the country code or prefixes	Yes
508	Mobile number is temporarily disabled	Yes
509	Phone is damaged	Yes
510	Phone is lost	Yes
511	The mobile number entered already exists and has been assigned to a different SMSC	Yes
512	Your mobile number and confirmation do not match. Please re-enter	Yes
513	Phone is already assigned	Yes
601	Current Token:	No
602	Please enter the one time password from one of your existing devices here	No
603	Invalid one time password	Yes
604	Invalid PAN	Yes
605	Device is not active	Yes

Authentication Device Messages		
606	Device is lost	Yes
607	Device is damaged	Yes
608	Device is already assigned	Yes
701	Email Token:	No
702	Please enter the one time password which was sent to you via Email	No
703	Invalid Email one time password	Yes
704	Invalid Email Address	Yes
705	Email is lost	Yes
706	Email is damaged	Yes
707	Your Email and confirmation do not match. Please re-enter	Yes
708	Email is already assigned	Yes
709	Unicode characters are not accepted	Yes

Local Pages Errors

Local Pages Errors	
Code	**Message**
101	Please re-enter the field(s) highlighted in red
102	Required field missing
103	Invalid number

Local Pages Errors	
104	Invalid SecureCode Invalid Verified by Visa Password Invalid JSecure Password Invalid SafeKey Invalid ProtectBuy Password
105	Invalid activation code
106	Data verification error
107	Field length exceeded
108	Invalid one time password
109	Invalid cardholder name
112	Your SecureCode and confirmation do not match. Please re-enter. Your Verified by Visa Password and confirmation do not match. Please re-enter. Your JSecure and confirmation do not match. Please re-enter Your SafeKey and confirmation do not match. Please re-enter. Your ProtectBuy Password and confirmation do not match. Please re-enter.
113	Invalid answer
114	Invalid username
115	Invalid full name
116	Invalid personal assurance message (PAM)
117	Invalid expiry date
118	Invalid card number
119	Invalid CVC
120	Invalid question
121	Invalid device type selected

Local Pages Errors	
122	Resynchronization failed
123	Invalid Password length Your SecureCode must be less "maxPassLen" characters long Your Verified by Visa Password must be less than "maxPassLen" characters long Your SecureCode must be less "maxPassLen" characters long Your Verified by Visa Password must be less than "maxPassLen" characters long Your JSecure Password must be less than "maxPassLen" characters long Your SafeKey must be less than "maxPassLen" characters long Your Password must be less than maxPassLen" characters long
124	Your SecureCode must be less "maxPassLen" characters long Your Verified by Visa Password must be less than "maxPassLen" characters long
125	Your SecureCode must contain at least "minPassDigit" digit(s) Your Verified by Visa must contain at least "minPassDigit" digit(s) JSecure must contain at least "minPassDigit" digit(s) SafeKey must contain at least "minPassDigit" digit(s) Password must contain at least "minPassDigit" digit(s)
126	Your SecureCode must contain at least "minPassCapital" capital letter(s) Your Verified by Visa must contain at least "minPassCapital" capital letter(s) Your JSecure must contain at least "minPassCapital" capital letter(s) Your SafeKey must contain at least "minPassCapital" capital letter(s) Your Password must contain at least "minPassCapital" capital letter(s)
127	Unicode characters are not accepted
128	Invalid character
129	Device is already assigned
130	Invalid PriSec
131	The Personal Message must not contain your Verified by Visa password or Password Hint
132	The Password Hint must not contain your Verified by Visa password
150	This field cannot be left blank

Local Pages Errors	
303	Invalid one time password
304	Invalid serial number
305	Device is lost
306	Device is damaged
307	Device is already assigned
403	Invalid one time password
404	Invalid serial number
405	Device is lost
406	Device is damaged
407	Device is already assigned
503	Invalid SMS one time password
504	Mobile number does not match the specified mobile restriction pattern
505	Invalid mobile network provider
506	Invalid country phone code
507	Please only enter mobile phone number without country code and prefixes
508	Mobile number has been temporarily disabled
509	Mobile phone for this number has been reported as damaged
510	Mobile phone for this number has been reported as lost
511	There is an already existing mobile number which has been assigned to a different SMSC

Local Pages Errors	
512	Your Mobile Number was not correctly confirmed. Please make sure that the Mobile Number and confirmation match
513	Phone is already assigned
603	Invalid one time password
604	Invalid PAN
605	Device is not active
607	Device is damaged
608	Device is already assigned

Transaction Status Codes

The following table lists the conditions that produce specific transaction status results.

Condition	Card Scheme	Status
Disabled Issuer	Visa, Mastercard, JCB, American Express, Diners Club	Ares.transStatus=R Ares.transStatusReason=12
Disabled Issuer Card Bin	Visa, Mastercard, JCB, American Express, Diners Club	Ares.transStatus=R Ares.transStatusReason=12
Issuer License invalid or expired	Mastercard	Ares.transStatus=N Ares.transStatusReason=12
Issuer License invalid	Visa, JCB, American Express, Diners Club	Ares.transStatus=U Ares.transStatusReason=12
Issuer License does not support 3DS2	Mastercard	Ares.transStatus=N Ares.transStatusReason=12
Issuer License does not support 3DS2	Visa, JCB, American Express, Diners Club	Ares.transStatus=U Ares.transStatusReason=12
Issuer License does not support the appBased device channel	Mastercard	Ares.transStatus=N Ares.transStatusReason=03
Issuer License does not support the appBased device channel	Visa, JCB, American Express, Diners Club	Ares.transStatus=U Ares.transStatusReason=03
Issuer License does not support the NPA message category	Mastercard	Ares.transStatus=N Ares.transStatusReason=20
Issuer License does not support the NPA message category	Visa, JCB, American Express, Diners Club	Ares.transStatus=U Ares.transStatusReason=20
Issuer License does not support the threeRI device channel	Mastercard	Ares.transStatus=N Ares.transStatusReason=21

Condition	Card Scheme	Status
Issuer License does not support the threeRI device channel	JCB, American Express, Diners Club	Ares.transStatus=U Ares.transStatusReason=21
Issuer License does not support the threeRI device channel	Visa	Ares.transStatus=U Ares.transStatusReason=12
purchaseDate is after now (+1 hour tolerance)	Visa, MasterCard, JCB, American Express, Diners Club	Ares.transStatus=N Ares.transStatusReason=07
Issuer License does not support RBA	Mastercard	Ares.transStatus=N Ares.transStatusReason=12
Issuer License does not support RBA	Visa, JCB, American Express, Diners Club	Ares.transStatus=U Ares.transStatusReason=12
CardExpiry < RecurringExpiry	Visa, Mastercard, JCB, American Express, Diners Club	Ares.transStatus=N Ares.transStatusReason=04
RecurringExpiry < PurchaseDate	Visa, Mastercard, JCB, American Express, Diners Club	Ares.transStatus=N Ares.transStatusReason=04
Areq.TDSCompId is Y and no ThreeDSMethodReq is received	Mastercard	Ares.transStatus=N Ares.transStatusReason=11
Areq.TDSCompId is Y and no ThreeDSMethodReq is received	Visa, JCB, American Express, Diners Club	Ares.transStatus=U Ares.transStatusReason=11
Areq.TDSCompId is U and ThreeDSMethodReq is received	Mastercard	Ares.transStatus=N Ares.transStatusReason=11
Areq.TDSCompId is U and ThreeDSMethodReq is received	Visa	Ares.transStatus=U Ares.transStatusReason=11
Browser data collected in ThreeDSMethodReq differs with Areq Browser Data	Mastercard	Ares.transStatus=N Ares.transStatusReason=11
Browser data collected in ThreeDSMethodReq differs with Areq Browser Data	Visa, JCB, American Express, Diners Club	Ares.transStatus=U Ares.transStatusReason=11

Condition	Card Scheme	Status
Card has been enrolled in the system (Pre-Registered) but not registered, and Proof of Authentication Attempt is enabled	Visa	Ares.transStatus= A Ares.transStatusReason=98
Card has been enrolled in the system (Pre-Registered) but not registered, and Proof of Authentication Attempt is enabled	Mastercard	Ares.transStatus= N Ares.transStatusReason=12
Card has been enrolled in the system (Pre-Registered) but not registered, and Proof of Authentication Attempt is enabled	JCB, American Express, Diners Club	Ares.transStatus= A Ares.transStatusReason=13
Card not enrolled in system and Proof of Authentication Attempt is enabled	Visa	Ares.transStatus= A Ares.transStatusReason=98
Card not enrolled in system and Proof of Authentication Attempt is enabled/disabled	Mastercard	Ares.transStatus= N Ares.transStatusReason=08
Card not enrolled in system and Proof of Authentication Attempt is enabled	JCB, American Express, Diners Club	Ares.transStatus=A Ares.transStatusReason=08
Card not enrolled in system and Proof of Authentication Attempt is disabled	Visa, Mastercard, JCB, American Express, Diners Club	Ares.transStatus=N Ares.transStatusReason=08
ACS records show the card type belongs to a provider that differs from the provider sent the request	Visa, Mastercard, JCB, American Express, Diners Club	Ares.transStatus= N Ares.transStatusReason=06
Device is listed in Unsupported Device lists	Mastercard	Ares.transStatus= N Ares.transStatusReason=03
Device is listed in Unsupported Device lists	Visa, JCB, American Express, Diners Club	Ares.transStatus= U Ares.transStatusReason=03
Error in parsing Device Info	Mastercard	Ares.transStatus= N Ares.transStatusReason=02
Error in parsing Device Info	Visa, JCB, American Express, Diners Club	Ares.transStatus= U Ares.transStatusReason=03

Condition	Card Scheme	Status
Challenge is decided for ThreeRI device channel	Visa, Mastercard, JCB, American Express, Diners Club	Ares.transStatus= N Ares.transStatusReason=15
No device is assigned to card	Mastercard	Ares.transStatus= N Ares.transStatusReason=12
No data available for card authentication	Visa, JCB, American Express, Diners Club	Ares.transStatus= U Ares.transStatusReason=12
ACS does not support SDK UI	Mastercard	Ares.transStatus= N Ares.transStatusReason=03
ACS does not support SDK UI	Visa, Mastercard, JCB, American Express, Diners Club	Ares.transStatus= U Ares.transStatusReason=03
ACS does not support SDK Interface	Mastercard	Ares.transStatus= N Ares.transStatusReason=03
ACS does not support SDK Interface	Visa, JCB, American Express, Diners Club	Ares.transStatus= U Ares.transStatusReason=03
Card status is set to stolen	Visa, Mastercard, JCB, American Express, Diners Club	Ares.transStatus= N Ares.transStatusReason=10
Card status is set to locked	Visa, Mastercard, JCB, American Express, Diners Club	Ares.transStatus= N Ares.transStatusReason=01
Card status is set to Disabled	Visa, Mastercard, JCB, American Express, Diners Club	Ares.transStatus= N Ares.transStatusReason=01
Card status is set to Deleted	Visa, Mastercard, JCB, American Express, Diners Club	Ares.transStatus= N Ares.transStatusReason=08

Condition	Card Scheme	Status
CardExpiry of Areq differs from real CardExpiry	Visa, Mastercard, JCB, American Express, Diners Club	Ares.transStatus= N Ares.transStatusReason=01
Card name of Areq differs from real card name	Mastercard	Ares.transStatus= N Ares.transStatusReason=12
Card name of Areq differs from real card name	Visa, JCB, American Express, Diners Club	Ares.transStatus= N Ares.transStatusReason=13
Card is expired	Visa, Mastercard, JCB, American Express, Diners Club	Ares.transStatus= N Ares.transStatusReason=05
Card is in reactivation mode	Mastercard	Ares.transStatus= N Ares.transStatusReason=15
Card is in reactivation mode	Visa, JCB, American Express, Diners Club	Ares.transStatus= N Ares.transStatusReason=15
In Authentication Exemption Settings, status for exempted transactions is defined Y SOFTLAUNCH LIST Enabled	Visa	Ares.transStatus= Y Ares.transStatusReason=99
In Authentication Exemption Settings, status for exempted transactions is defined A SOFTLAUNCH LIST Enabled	Visa	Ares.transStatus= A Ares.transStatusReason=99
In Authentication Exemption Settings, status for exempted transactions is defined A or Y SOFTLAUNCH LIST Enabled	Mastercard	Ares.transStatus= Y Ares.transStatusReason=99
In Authentication Exemption Settings, status for exempted transactions is defined Y SOFTLAUNCH LIST Enabled	JCB, American Express, Diners Club	Ares.transStatus= Y Ares.transStatusReason=17
In Authentication Exemption Settings, status for exempted transactions is defined A SOFTLAUNCH LIST Enabled	JCB, American Express, Diners Club	Ares.transStatus= A res.transStatusReason=17

Condition	Card Scheme	Status
In Authentication Exemption Settings, status for exempted transactions is defined Y Domestic and International Amount threshold Enabled	Visa	Ares.transStatus= Y Ares.transStatusReason=99
In Authentication Exemption Settings, status for exempted transactions is defined A Domestic and International Amount threshold Enabled	Visa	Ares.transStatus= A Ares.transStatusReason=99
In Authentication Exemption Settings, status for exempted transactions is defined A or Y Domestic and International Amount Threshold Enabled	Mastercard	Ares.transStatus= Y Ares.transStatusReason=99
In Authentication Exemption Settings, status for exempted transactions is defined Y Domestic and International Amount Threshold Enabled	JCB, American Express, Diners Club	Ares.transStatus= Y Ares.transStatusReason=17
In Authentication Exemption Settings, status for exempted transactions is defined A Domestic and International Amount Threshold Enabled	JCB, American Express, Diners Club	Ares.transStatus= A Ares.transStatusReason=17
Risk Chain is Enabled	Visa	Ares.transStatus= Y Ares.transStatusReason=99
Risk Chain is Enabled	Mastercard, JCB, American Express, Diners Club	Ares.transStatus= Y Ares.transStatusReason=17
Certificate is not found or expired	Mastercard	Ares.transStatus= N Ares.transStatusReason=09
Certificate is not found or expired	Visa, JCB, American Express, Diners Club	Ares.transStatus= U Ares.transStatusReason=09
Challenge is required	Visa, Mastercard, JCB, American Express, Diners Club	Ares.transStatus= C

RReq Authentication Method Codes

The following table lists the possible action/results that may be produced by each RReq Authentication Method.

RREQ.AUTHENTICATIONMETHOD	Condition	Action/Results
01 = STATIC PASSCODE	Card has any static authData such as password, MRN, etc, and no device	- Timeout occurs - Completion of the authentication - Cancel on authentication page
02 = SMS OTP	Card has SMS and optionally an authData like password	- Timeout occurs - Completion of the authentication - Cancel on OTP page
03 = KEY FOB OR EMV CARD READER OTP	Not supported	
04 = APP OTP	Remote cards only: AuthType=4 (Virtual OTP device)	- Timeout occurs - Completion of the authentication - Cancel on OTP page
05 = OTP OTHER	All dynamic authentication such as email, Vasco, C&R, etc	- Timeout occurs - Completion of the authentication - Cancel on OTP page
06 = KBA	Not supported	
07 = OOB BIOMETRICS	Remote cards only: AuthType=6 (Verified by Voice) and AuthType=11 (BIO)	- Timeout occurs - Completion of the authentication - Cancel on OOB page

RREQ.AUTHENTICATIONMETHOD	Condition	Action/Results
08 = OOB LOGIN	Remote cards only: AuthType=9 (Online Banking)	- Timeout occurs - Completion of the authentication - Cancel on OOB page
09 = OOB OTHER	OOB authentication	- Timeout occurs - Completion of the authentication - Cancel on OOB page
11 = PUSH CONFIRMATION	Decoupled authentication	- Timeout occurs - Completion of the authentication

Glossary

This page provides a list of terms relating to 3D Secure 1 and 2, some are not used elsewhere in this documentation but are included for completeness of the subject area. Familiarise yourself with them now or refer back to this page when you come across an unfamiliar word, phrase or acronym.

Term	Acronym	Definition
2-F Authentication		A generic functionality, which allows for strong authentication of any transaction, commercial or otherwise, for example, strong authentication of users when they login to an Internet banking site or when they authorise funds transfer to a third party. 2-F authentication requires two independent ways to establish identity and privileges as opposed to traditional password authentication, which requires only one 'factor' (knowledge of a password).
3-D Secure **3D Secure** **3D Secure 1** **3D Secure 2**	**3DS** **3DS1** **3DS2**	A payer authentication standard (3D Secure 1 (3DS1)) introduced by Visa (Verified by Visa) and subsequently adopted by Mastercard (Mastercard SecureCode and Mastercard SecureCode), JCB (JCB J/Secure), American Express (SafeKey) and Diners Club International / Discover (ProtectBuy) designed to reduce online credit card fraud and chargeback. The 3DS standard provides an additional layer of protection in card-not-present credit card transactions for the three domains involved: Issuer domain of the card issuing bank, the Interoperability domain of the card scheme's infrastructure and the Acquirer domain of the merchants. The second version of the standard, 3D Secure 2 (3DS2) (EMV 3-D Secure protocol), is facilitated by EMVCo, a six member consortium comprised of American Express, Discover, JCB, Mastercard, UnionPay and Visa. It creates a frictionless payment experience for cardholders by facilitating a richer cardholder data exchange, allowing risk-based authentication by issuers for low risk transactions, instead of authentication challenges to the cardholder, such that most authentication activity will be invisible to the cardholder. 3DS2 also supports authentication of app-based transactions on mobile and other consumer connected devices, and cardholder verification for non-payment transactions, such as adding a payment card to a digital wallet.

Term	Acronym	Definition
3DS Client		The consumer-facing component, such as a browser-based or mobile app online shopping site, which facilitates consumer interaction with the 3DS Requestor for initiation of the EMV 3-D Secure protocol.
3DS Integrator		An EMV 3-D Secure participant that facilitates and integrates the 3DS Requestor Environment, and optionally facilitates integration between the Merchant and the Acquirer.
3-D Secure Provider		An entity such as American Express, Diners Club International, Discover, JCB, Mastercard or Visa, which provides interoperability services for issuers and merchants who participate in the authentication process. The 3-D Secure provider is normally in charge of managing the directory server, managing the authentication history server and issuing the digital certificates required for participation in the authentication scheme.
3DS Requestor		The initiator of the EMV 3-D Secure Authentication Request, known as the AReq message. For example, this may be a merchant or a digital wallet requesting authentication within a purchase flow.
3DS Requestor App		An App on a Consumer Device that can process a 3-D Secure transaction through the use of a 3DS SDK. The 3DS Requestor App is enabled through integration with the 3DS SDK.
3DS Requestor Environment		This describes the 3DS Requestor controlled components of the Merchant / Acquirer domain, which are typically facilitated by the 3DS Integrator. These components include the 3DS Requestor App, 3DS SDK, and 3DS Server. Implementation of the 3DS Requestor Environment will vary as defined by the 3DS Integrator.
Three Domain Secure Software Development Kit	**3DS SDK**	3-D Secure Software Development Kit. A component that is incorporated into the 3DS Requestor App. The 3DS SDK performs functions related to 3-D Secure on behalf of the 3DS Server.
3DS Requestor Initiated	**3RI**	3-D Secure transaction initiated by the 3DS Requestor for the purpose of confirming an account is still valid. The main use case being recurrent transactions (TV subscriptions, utility bill payments, etc.) where the merchant wants perform a Non-Payment transaction to verify that a subscription user still has a valid form of payment.
3DS Server		Refers to the 3DS Integrator's server or systems that handle online transactions and facilitate communication between the 3DS Requestor and the Directory Server.

Term	Acronym	Definition
3-D Secure	**3DS**	**Three Domain Secure**. An eCommerce authentication protocol that for version 2 onwards enables the secure processing of payment, non-payment and account confirmation card transactions.
Access Control Server	**ACS**	A component that operates in the Issuer Domain, which verifies whether authentication is available for a card number and device type, and authenticates specific Cardholders.
Accountholder Authentication Value	**AAV**	A value providing proof of cardholder authentication, which is generated by the issuer's access control server for each transaction. The AAV is passed by the merchant to the acquirer and then by the acquirer to the issuer through the UCAF field.
Acquirer		A financial institution that has a relationship with a merchant and processes payment transactions for that merchant.
ActiveAccess		GPayments' access control server for card issuers and service providers.
ActiveDevice		GPayments' device agnostic two-factor authentication component.
ActiveMerchant		GPayments' payment authentication platform (merchant plug-in) for merchants.
ActiveServer		GPayments' 3DS Server for payment processors and merchants (see *3DS Server*).
Attempts		Used in the EMV 3DS specification to indicate the process by which proof of an authentication attempt is generated when payment authentication is not available. Support for Attempts is determined by each DS.
Authentication		In the context of 3-D Secure, the process of confirming that the person making an eCcommerce transaction is entitled to use the payment card.
Authentication Device		A physical device capable of generating a token to be used in the verification of a user's identity.
Authentication Request Message	**AReq**	An EMV 3-D Secure message sent by the 3DS Server, via the DS, to the ACS to initiate the authentication process.

Term	Acronym	Definition
Authentication Response Message	**ARes**	An EMV 3-D Secure message returned by the ACS, via the DS, in response to an Authentication Request message.
Authentication Token		An unpredictable piece of information generated by an authentication device, which is used to verify the identity of a user. The term token may sometimes be used to refer to the physical device that generated the token as well.
Authentication Value	**AV**	A cryptographic value generated by the ACS to provide a way, during authorisation processing, for the authorisation system to validate the integrity of the authentication result. The AV algorithm is defined by each Payment System.
Authorisation		A process by which an Issuer, or a processor on the Issuer's behalf, approves a transaction for payment.
Authorisation System		The systems and services through which a Payment System delivers online financial processing, authorisation, clearing, and settlement services to Issuers and Acquirers.
Bank Identification Number	**BIN**	The first six digits of a payment card account number that uniquely identifies the issuing financial institution. Also referred to as an Issuer Identification Number (IIN) in ISO 7812.
BankNet		Mastercard's proprietary payment network.
Base64		Encoding applied to the Authentication Value data element as defined in RFC 2045.
Base64 URL		Encoding applied to the 3DS Method Data, Device Information and the CReq/CRes messages as defined in RFC 7515.
Card		Card is synonymous with the account of a payment card, in the *EMV 3-D Secure Protocol and Core Functions Specification*.
Certificate Authority	**CA**	
Cardholder		An individual to whom a card is issued or who is authorised to use that card.

Term	Acronym	Definition
Cardholder Activation During Shopping		A 3D-Secure 1 process by which cardholders can enrol with the authentication system at the time of making a purchase at a participating merchant eCommerce website.
Centralised Authentication and Authorisation Service	**CAAS**	A remote ACS, see *Access Control Server*.
Challenge		The process where the ACS is in communication with the 3DS Client to obtain additional information through Cardholder interaction.
Challenge Flow		A 3-D Secure flow that involves Cardholder interaction as defined in the *EMV 3-D Secure Protocol and Core Functions Specification*.
Challenge Request Message	**CReq**	An EMV 3-D Secure message sent by the 3DS SDK or 3DS Server where additional information is sent from the Cardholder to the ACS to support the authentication process.
Challenge Response Message	**CRes**	The ACS response to the CReq message. It can indicate the result of the Cardholder authentication or, in the case of an App-based model, also signal that further Cardholder interaction is required to complete the authentication.
Chip Card		A card with an on-board integrated circuit chip.
Consumer Device		Device used by a Cardholder such as a smartphone, laptop, or tablet that the Cardholder uses to conduct payment activities including authentication and purchase.
Cryptography		A process that encrypts information for the purpose of protecting it. Information is decrypted when required.
Device		see *Authentication Device*.
Device Channel		Indicates the channel from which the transaction originated. Either: • App-based (01-APP) • Browser-based (02-BRW) • 3DS Requestor Initiated (03-3RI)
Device Information		Data provided by the Consumer Device that is used in the authentication process.

Term	Acronym	Definition
Directory Server	**DS**	A server component operated in the Interoperability Domain; it performs a number of functions that include: authenticating the 3DS Server, routing messages between the 3DS Server and the ACS, and validating the 3DS Server, the 3DS SDK, and the 3DS Requestor.
Directory Server Certificate Authority	**DS CA** or **CA DS**	A component that operates in the Interoperability Domain; generates and Certificate Authority (DS distributes selected digital certificates to components participating in 3-D Secure. Typically, the Payment System to which the DS is connected operates the CA.
Directory Server ID (directoryServerID)		Registered Application Provider Identifier (RID) that is unique to the Payment System. RIDs are defined by the ISO 7816-5 standard.
Electronic Commerce Indicator	**ECI**	Payment System-specific value provided by the ACS to indicate the results of the attempt to authenticate the Cardholder.
Digital Signature		Equivalent of the physical signature in the digital world. Digital signatures can verify the identity of owner of a piece of information or a document in the digital world.
Enrolment		A cardholder must pass an initial online authentication procedure in 3D-Secure 1, which is verified by the Issuer prior to gaining eligibility for participation in American Express SafeKey, Diners Club International ProtectBuy, JCB J/Secure, Mastercard SecureCode or Verified by Visa authentication.
Frictionless		Used to describe the authentication process when it is achieved without Cardholder interaction.
Frictionless Flow		A 3-D Secure flow that does not involve Cardholder interaction as defined in EMVCo Core Spec Section 2.5.1.
Issuer		A financial institution that provides cardholders with credit cards.
J/Secure		JCB's standard for cardholder authentication, based on 3-D Secure.
Message Authentication Code	**MAC**	

Term	Acronym	Definition
Mastercard SecureCode / Identity Check		Mastercard's payer authentication brand, which includes SPA Algorithm for the Mastercard Implementation of 3-D Secure, SPA and chip card authentication program (CAP).
Mastercard 3-D Secure		The SPA Algorithm for the Mastercard Implementation of 3-D Secure that provides a browser authentication experience to the cardholder (see also *3-D Secure)*.
Mastercard Identity Check		see *Mastercard SecureCode / Identity Check*.
Merchant		Entity that contracts with an Acquirer to accept payments made using payment cards. Merchants manage the Cardholder online shopping experience by obtaining the card number and then transfers control to the 3DS Server, which conducts payment authentication.
Merchant Plug-in (MPI)		A software module which can be integrated into a merchant's eCommerce website or run as a managed service on behalf of a number of merchants to provide 3-D Secure authentication.
Non-Payment Authentication	**NPA**	.
One-Time Passcode	**OTP**	A passcode that is valid for one login session or transaction only, on a computer system or other digital device.
Out-of-Band	**OOB**	A Challenge activity that is completed outside of, but in parallel to, the 3-D Secure flow. The final Challenge Request is not used to carry the data to be checked by the ACS but signals only that the authentication has been completed. ACS authentication methods or implementations are not defined by the 3-D Secure specification.
Payer Authentication Request	**PAReq**	Message sent from the MPI to the Access Control Server at the cardholder's issuer via the cardholder browser.
Payer Authentication Response	**PARes**	A digitally signed message sent from the Access Control Server to the Merchant Plug-in which communicates whether the cardholder authentication was successful or not.

Term	Acronym	Definition
Payment Gateway		A software system provided by an acquirer or a third party which accepts transactions from the Internet and transfers them to a payment network such as BankNet or VisaNet.
Preparation Request Message	**PReq**	3-D Secure message sent from the 3DS Server to the DS to request the ACS and DS Protocol Versions that correspond to the DS card ranges as well as an optional 3DS Method URL to update the 3DS Server's internal storage information.
Preparation Response Message	**PRes**	Response to the PReq message that contains the DS Card Ranges, active Protocol Versions for the ACS and DS and 3DS Method URL so that updates can be made to the 3DS Server's internal storage.
Proof or authentication attempt		Refer to Attempts.
ProtectBuy		Diners Club International and Discover standard for cardholder authentication, based on 3-D Secure.
Registered Application Provider Identifier	**RID**	Registered Application Provider Identifier (RID) is unique to a Payment System. RIDs are defined by the ISO 7816-5 Standard and are issued by the ISO/IEC 7816-5 Registration Authority. RIDs are 5 bytes.
Results Request Message	**RReq**	Message sent by the ACS via the DS to transmit the results of the authentication transaction to the 3DS Server.
Results Response Message	**RRes**	Message sent by the 3DS Server to the ACS via the DS to acknowledge receipt of the Results Request message.
Risk-Based Authentication	**RBA**	During risk-based authentication, the rich cardholder data exchanged in AReq is taken into account to determine the risk profile associated with that transaction. The complexity of the challenge is then decided based on the risk profile.
SafeKey		American Express standard for cardholder authentication, based on 3-D Secure.

Term	Acronym	Definition
Secure Payment Application (SPA)		Mastercard's payer authentication standard designed to reduce online credit card fraud and chargeback using a client-side applet. Also known as Mastercard's PC Authentication Program, Mastercard SecureCode, Mastercard SPA and SPA.
Secure Sockets Layer (SSL)		A protocol designed to maintain the integrity and confidentiality of communication over the Internet.
SecureCode		see *Mastercard SecureCode / Identity Check*.
Token:		see *Authentication Token*.
Two Factor Authentication		see *2-F Authentication*
Uniform Resource Locator (URL)		Address system for locating unique sites on the Internet.
Universal Cardholder Authentication Field (UCAF)		Data element 48 sub element 43 as defined in Mastercard BankNet to carry authentication data. Mastercard SecureCode uses this element to transport AAV from the acquirer to the issuer.
Verified by Visa	**VbV**	A payer authentication standard introduced by Visa (see *3-D Secure*).
VisaNet		Visa's proprietary payment network.
Visa Secure		A program developed by Visa to make online payments more secure through 3-D Secure 2.

Document Control

new item item changed item removed no change to item

Date	AA Ver	Doc Ver	Change Details
[06/09/2021]	**9.0.4**	**9.0.4:1**	**Installation** (Installation Guide) Added new configuration parameters **IGNORE_DTD_ORDER_3DS1** and **PURCHASE_DATE_ACCEPT_BALANCE** in Common Configuration Parameter
			Decoupled Authentication Adapter (Specifications) Added new page: Decoupled Authentication Adapter.
[16/08/2021]	**9.0.3**	**9.0.3:1**	**Installation** (Installation Guide) Added new steps in Pre Installation Configurations and New Installations
			Issuers (Admin UI) Added new section: Import Key.
			Whitelisting (Specifications) Added new page: Whitelisting.
[30/07/2021]	**9.0.0**	**9.0.0:1**	**Product Architecture** (Installation Guide) Added **Payer Authentication Server**, **Decoupled Authentication Adapter** **Whitelisting Server** in Internal Components Changed **Authentication Server (3DS2)** and **Challenge Server (3DS2)** in Components Changed Logical view of ActiveAccess diagram.
			External Components (Installation Guide) Changed the SQL commands in Find Transactions Performance Added **OOB** and **Decoupled Authentication** in Two-Factor Authentication

Date	AA Ver	Doc Ver	Change Details
			Installation (Installation Guide) Removed **AA_Administration**, **MIA_DB_DESede** key aliases, and **the issue alias (e.g.** `Card< Issuer_ID >` **)** in Prerequisites and Installation of Individua Components Added new step for removing **enrolment.war** in Upgrades Added **Whitelisting** in Deploying WAR packages and Installation of Indivi Components Added an **important notice** in Post Installation Added Whitelist Server Removed **Module** in Additional Administration Server Configuration Para
			About the Issuer Administration Server (Admin UI) Added **Note** in Login.
			Settings (Admin UI) Added **Whitelisting server URL** in Settings.
			System Management (Admin UI) Added **Visa CEMEA region** in New Issuer Group, Issuer Group Details Added notes in Issuer Details Changed notes in Issuer Details Added **Whitelisting** in BIN Management Added **Decoupled Authenticator** in Edit Device Parameters.
			About Authentication Management (Admin UI) Added **Decoupled Authenticator Management** to **About Authentication Management**.
			Device Management (Admin UI) Removed all instances of **CAP** and **RSA** device types.
			Risk Management (Admin UI) Added **Adapter ID** and **Generate** in Register Risk Adapter.
			OOB Management (Admin UI) Added **Adapter ID** and **Generate** in Register / Edit OOB Adapter.
			Decoupled Authenticator Management (Admin UI) Added new page: Decoupled Authenticator Management.

Date	AA Ver	Doc Ver	Change Details
			Security (Admin UI) Added new section: Decoupled Authenticator Certificate.
			Migrate to Data Key Utility (Admin UI) Added new page: Migrate to Data Key Utility.
			Issuers (Admin UI) Added **Name on card verification** in Local Issuer Settings Added **ACS Reference Number** in Provider Settings Added displayed key details for **General keys** in Key Management Changed **Info** in New Key.
			Cards (Admin UI) Added **Decoupled Authenticator** in Find Card Added **Whitelisting** in Cards Details Added new section: Whitelisting Removed sections: **Find User** and **New User** in Cards.
			Transactions (Admin UI) Changed **3-D Secure version** in Search Fields Added **Frictionless by review** in Transaction Details Removed section: **Find ActiveDevice** in Transactions.
			Reports (Admin UI) Added **Decoupled Authenticator** in Card Summary Added new section: Device Summary Removed sections: **User Summary**, **User Authentication**, **User Activity** ar **Enrolment Activity** in Reports.
			SMS via JMS Messaging (Specifications) Changed **Tag** of **message_payload** in Optional Parameters.
			Out of Band (OOB) Authentication Adapter (Specifications) Added **threeDSRequestorAppURL**, **callbackUrl** and **instruction** to Out of E (OOB) Authentication Adapter.
			Codes (Transaction Status Codes) Added new conditions in Transaction Status Codes.

Date	AA Ver	Doc Ver	Change Details
			Codes (RReq Authentication Method Codes) Added **11 = PUSH CONFIRMATION** in RReq Authentication Method Code
			Removed all instances of **ActiveDevice/User Authentication** Removed all instances of **Enrolment Server** Removed all instances of **CAP** and **RSA** device types.
[17/06/2021]	**8.5.8**	**8.5.8:1**	**External Components** (Installation Guide) Added note regarding Tomcat 7 in Application Server
			Installation (Installation Guide) Added new configuration parameter **AMOUNT_FORMATTER** in Common Configuration Parameters
			Issuers (Admin UI) Added note in Upload Registration Files
			Cards (Admin UI) Added note in New SMS Device
			Local Messaging (Specifications) Added acceptable values for **Status** in Update Registration Request.
[30/04/2021]	**8.5.6**	**8.5.6:1**	**SMS via JMS Messaging** (Specifications) Added **Tag** and **Size** columns in Optional Parameters Changed all values in **Type** column in Optional Parameters Added new field names in Optional Parameters Added examples for **CLIENTID** in Examples.
[05/02/2021]	**8.5.3**	**8.5.3:1**	**Installation** (Installation Guide) Added **WS_POOL** to ActiveAccess Configuration File.
18/12/2020	**8.5.0**	**8.5.0:1**	**Product Architecture** (Installation Guide) Added **Oracle WebLogic Server 14c** and **Database Oracle 19c** in Hardwa Software Requirements.
			External Components (Installation Guide) Added additional steps for Oracle 19c in Oracle Database.

Date	AA Ver	Doc Ver	Change Details
			Installation (Installation Guide) Changes made to Prerequisites Added installation steps for Upgrades to v8.5.x and later Added new configuration parameters **MASTER_HSM_LIB_DIR** and **MASTER_HSM_SLOT** to Common Configuration Parameters Changes made to Installation of Individual Components.
			Risk Management (Admin UI) Added details about authentication method and **Score range for frictionl review** in Add/Edit Risk Chain.
			Servers (Admin UI) Added new section Edit ACS Server.
			Key Retiring Utility (Admin UI) Changes made to Retiring keys automatically.
			Issuers (Admin UI) Changes made to the description of Key Management Added **Export**, **KeyStore type** and a note for **Delete** to Key Management Added **HMAC keys** and an **Info** box to New Key Added **Export** and **KeyStore type** to Key Management Added **KeyStore type** to Key Details Removed **New Key** link from Key Details Added new section Export Data Key.
			Remote Messaging (Specifications) Added **purchaseDate** to **OobInfo** in Table 14 - InitAuthReq Added item 6 to **Code** in Table 17 - VerifyAuth.
			Out of Band (OOB) Authentication Adapter (Specifications) Added **NOT_AUTHENTICATED_END** to OobAuthenticationResult Data Ele
			Risk Engine Adapter (Specifications) Added **frictionless with review** in How RBA works.
[25/11/2020]	**8.4.4**	**8.4.4:1**	**Remote Messaging** (Specifications) Added **Attributes** and **Descriptions** to Table 4 - Transaction, Table 10 - PreAuthReq, Table 11 - HeaderParams and Table 12 - AdditionalParams.

Date	AA Ver	Doc Ver	Change Details
[29/10/2020]	8.4.1	8.4.1:1	**System Management** (ACS URL) Added details to **ACS challenge URL** for OOB's WebSocket and callback U 3-D Secure 2 Settings.
			System Management (Issuer Management) Added a note to **ACS Challenge URL** for OOB's WebSocket and callback U New Issuer Group, Issuer Group Details and Issuer Details.
			Remote Messaging (Specifications) Added notes to AuthType and AuthTypeSup at Table 6 - CardInfo.
			Codes (RReq Authentication Method Codes) Added new page: RReq Authentication Method Codes.
[16/10/2020]	8.4.0	8.4.0:1	**Settings** (Admin UI) Added a note to **Log level** in Settings.
			Issuer Management (Admin UI) Added **Verified by Visa CAVV format** and **Visa Secure CAVV format** to N Issuer Group and Issuer Details Added **IAV generation algorithm**, **Verified by Visa CAVV format** and **Visa CAVV format** to Issuer Group Details Added **ACS URL** to New Issuer.
			Issuers (Admin UI) Added a note to Custom pages.
			Transactions (Admin UI) Added **Failed reason** and **IAV generation algorithm** to Transaction Detail
			Remote Messaging (Specifications) Added **callBack** to Table 14 - InitAuthReq.
			Out of Band (OOB) Authentication Adapter (Specifications) Added **purchaseDate** to TransactionInfo Data Elements.
29/05/2020	8.3.0	8.3.0:1	**Installation** (Installation Guide) Added an option to change **RMI port** in Additional Administration Server Configuration Parameters.

Date	AA Ver	Doc Ver	Change Details
			Issuer Management (Admin UI) Added **IAV generation algorithm** to New Issuer Group Added a warning to **Supported devices** in ActiveDevice Settings.
			Device Management (Admin UI) Added **OOB** to Edit Default Device Parameters and OOB.
			Risk Management (Admin UI) Added Upload Connector Encryption Key.
			OOB Management (Admin UI) Added Upload Connector Encryption Key.
			Cards (Admin UI) Added **Deactivated device type** to **Status** in Assigned Devices.
			CardLoader (Specifications) Added encryption of sensitive data to **Log directory** in Open dialog for se XML file to verify.
			Remote Messaging (Specifications) Added **acsTransId**, **threeDSTransId** and **dsTransId** to Table 4 - Transactic
			Out of Band (OOB) Authentication Adapter (Specifications) Added samples to Get OOB Adapter Information and Request OOB Challe Added new **Length** for `acctNumber` in TransactionInfo Data Elements an `cardholderName` in CardHolderInfo Data Elements Added **Message Inclusion** for `clientId` and `deviceId` in AdditionalInfo Elements.
			Risk Engine Adapter (Specifications) Added **AReqWithTransStatus Data Elements** Added new **Length** for `acctNumber` and `cardholderName` in AReq Data E Added **Message Inclusion** for `clientId` in AdditionalInfo Data Elements

Date	AA Ver	Doc Ver	Change Details
24/04/2020	**8.2.3**	**8.2.3:1**	**Risk Engine Adapter** (Specifications) Changes made to **Parameter Data Elements** Change made to **Condition Data Elements** Added **ValueType Data Elements**, **ConditionAssessor Data Elements**, and **TxCallback Data Elements** Changes made to **ConditionValue Data Elements** Added **Range Data Elements** Change made to **messageExtension Data Elements** Removed **AdapterRiskAssessmentOutput Data Elements**.
17/4/2020	**8.2.0**	**8.2.0:2**	**Remote Messaging** (Specifications) Added attribute lengths to the **Usage** column of Table 2 - VerifyRegReq, T Card, Table 4 - Transaction and Table 14 - InitAuthReq.
			Out of Band (OOB) Authentication Adapter (Specifications) Changes made to **Out of Band (OOB) Authentication Adapter** (Specificat
28/02/2020	**8.2.0**	**8.2.0:1**	**Installation** (Installation Guide) Added **TOMCAT_KEYSTORE**, **TOMCAT_KEYSTORE_PASS**, **TOMCAT_TRUSTSTORE** and **TOMCAT_TRUSTSTORE_PASS** to configuratior
			Issuer Management (Admin UI) Added **IAV generation algorithm** to Issuer Details.
			Risk Management (Admin UI) Change made to **Score range for device** in Add / Edit Risk Chain.
			Servers (Admin UI) Added **OOB info template** to Edit CAAS Server.
			Issuers (Admin UI) Added **Maximum interaction** to Remote Issuer Settings.
			Cards (Admin UI) Added **Client ID** to Find Card and Card Details Added note to **Expiry date** in New Card and Card Details.

Date	AA Ver	Doc Ver	Change Details
			Transactions (Admin UI) ➕ Added **Client ID** to Find 3-D Secure ➕ Added **Risk decision** and **Client ID** to Transaction Details.
			Local Messaging (Specifications) ➕ 🔺 Additions & changes made for **Client ID** to: 🔺 Sample pre-registration request 🔺 Sample final registration request for traditional 3-D Secure 🔺 Sample final registration request for two-factor authentication over 3-D S 🔺 Sample update registration request ➕ Card Device Update Request 🔺 Sample Registration Notification 🔺 Sample Device Update Notification 🔺 Sample Opt-Out Notification 🔺 Sample Lock Notification 🔺 Cardholder Registration DTD.
			Remote Messaging (Specifications) ➕ Added **LanCode** to Table 3 - Card and Table 6 - CardInfo ➕ Added **twoFA** to Table 6 - CardInfo.
			Out of Band (OOB) Authentication Adapter (Specifications) 🔺 Changes made to Adapter Interface Methods 🔺 Change made to **Response Description** of Get OOB Adapter information 🔺 Change made to **Response Description** of Request OOB Challenge 🔺 Change made to **Request Method** and **Response Description** of Get OOB authentication result ➕ Added AdapterInfo Data Elements 🔺 Change made to `acctNumber` **Description** in TransactionInfo Data Eleme ➕ Added `deviceId` to AdditionalInfo Data Elements ➕ OobRequestChallengeResult Data Elements added ➕ OobAuthenticationResult Data Elements added.
10/01/2020	**8.1.2**	**8.1.2:1**	**Installation** (Installation Guide) ➕ Added **JSON Response Elements** in ACS, MIA, Registration and Enrolmer
			Profile Management (Admin UI) 🔺 Change made to **2-factor authentication** login option in User Profile.

Date	AA Ver	Doc Ver	Change Details
			Remote Messaging (Specifications) Change made to **Description** and **Sample Value** of **AuthType** in Pre Authentication Response.
			Local Messaging (Specifications) Changes made to **Request** and **Response** of **Cardholder Registration** Changes made to **Request** and **Response** of **Notification** Changes made to **Critical Card Data Encryption and Decryption** Changes made to **Cardholder Registration** Changes made to **Notification**.
06/12/2019	**8.1.1**	**8.1.1:1**	**Installation** (Installation Guide) Added monitoring of the availability of ACS, MIA, Registration and Enrolm
			Device Management (Admin UI) Added `Plus (+) prefix` in SMS Center.
			Issuers (Admin UI) Change made to Language selection during authentication: add *authenti process of 3-D Secure 1* Change made to Provider Settings: add JSON format examples.
			Local Messaging (Specifications) Change made to Request: Update EncVectorIV Update Sample final registration request for traditional 3-D Secure Change made to Cancel Registration Request: Make name attribute of ca optional Change made to Critical Card Data Encryption and Decryption: Change ke algorithm to AES Change made to Cardholder Registration DTD: Change `Name CDATA` to `I`
			Out of Band (OOB) Authentication Adapter (Specifications) Added `Swagger API URL` to Restful API version of OOB Adapter.
			Risk Engine Adapter (Specifications) Added `Swagger API URL` to RESTful API Risk Adapter.
			Codes (Error Codes) Added Error codes to Server Error Codes.

Date	AA Ver	Doc Ver	Change Details
15/11/2019	**8.1.0**	**8.1.0:1**	**Installation** (Installation Guide) Removed **HSM_LIB_DIR** parameter from Upgrades to v8.x.x.
			System Management (Admin UI) Change made to New Issuer Group, Issuer Group Details, and Issuer Deta Changes `MAC Algorithm` to **3DS1 only** and changed `Use parent certific` `public and encryption keys`. Change made to Public & Encryption Key Management: Change key algo **AES**.
			Security (Admin UI) Added new section: SDK certificate.
			Cards (Admin UI) Change made to New Card: The card Expiry date is mandatory for Maste
			Risk Engine Adapter (Specifications) Removed one method of `TxCallback` from Parameter Data Elements. Removed `resultWhenTransmissionError` from RemoteCondition Data E Added `range` field into ConditionValue Data Elements
06/11/2019	**8.0.3**	**8.0.3:1**	**Risk Engine Adapter** (Specifications) Change made to AdapterInfo Data Elements: Removed round brackets fr **Token** Sample Value. Change made to AssessmentResult Data Elements: Change the descripti `whatToDoNext` `range` field added into ConditionValue Data Elements
09/10/2019	**8.0.2**	**8.0.2:2**	**Remote Messaging** (Specifications) Change made to Table 16 - VerifyAuthReq: Removed round brackets from **Token** Sample Value.
			Out of Band (OOB) Authentication Adapter (Specifications) Change made to **oobAuthenticationResult**: Add `PENDING` as a valid value
			Risk Engine Adapter (Specifications) Changed **Risk chain setup diagram**.

Date	AA Ver	Doc Ver	Change Details
02/10/2019	8.0.2	8.0.2:1	**Installation** (Installation Guide) Changes made to Upgrades to v8.x.x: Addition of **HSM_LIB_DIR** paramet updates to JAR files which must be removed. Addition of **HSM_LIB_DIR**, **HSM_SLOT**, **TESTING_MODE**, **PROVIDER_TES TEST_AUTH_SERVER**, and **ACS_REFERENCE_NUMBER_TEST** to Common configuration parameters.
			Remote Messaging (Specifications) Added **Response code = 3**.
			Codes (Transaction Status Codes) Added new page: Transaction Status Codes.
05/09/2019	8.0.1	8.0.1:1	**Product Architecture** (Installation Guide) Added Disaster Recovery and Clustering diagrams.
			Installation (Installation Guide) Changes made to Upgrades to v8.0.x and New installations.
			Security (Admin UI) Added new **Key type** field to Create Certificate Request.
			Risk Engine Adapter (Specifications) Changed **Validator** field description in **ParameterDataElements** Chenged **PreviousData** field format in **RemoteAssessmentRequest Data Elements** Added **AReqWithTransStatusDataElements** Changed **ThreeDSCompInd** and **ThreeDSRequestorAuthenticationInd** fie **AReq Data Elements**.
			Remote Messaging (Specifications) **InitAuthReq** table: Usage of **oobInfo** changed.
			Out of Band (OOB) Authentication Adapter (Specifications) Change the URL in **Restful API version of OOB Adapter** Change `NOT__AUTHENTICATED` to `NOT_AUTHENTICATED` Update MobilePhone Data Elements, HomePhone Data Elements, and W Data Elements.

Date	AA Ver	Doc Ver	Change Details
15/08/2019	**8.0.0**	**8.0.0:1**	**Product Architecture** (Installation Guide) Components labelled with (3DS1) or (3DS2) as relevant Added Challenge Server (3DS2). Added Risk Engine Adapter Added Out of Band (OOB) Authentication Adapter Changed Logical view of ActiveAccess diagram Changed Hardware and Software Requirements Removed references to **RuPay** components.
			External Components (Installation Guide) Application Server dependency removed, supports compatible Java Appl Servers.
			Installation (Installation Guide) ActiveAccess installation and setup process simplified.
			System Management (Admin UI) **Authentication Management** section added with tabs for: Device Management previously under **System Management** Risk Management for 3DS2 risk management OOB Management for OOB processing support.
			System Management (Admin UI) - **Issuer Management** Device Settings: Added **OOB** as a supported device.
			Security (Admin UI) Added Directory Server Certificate section Added OOB Certificate section Added Risk Certificate section.
			Issuers (Admin UI) Providers parameters moved to a new page, and linked, from the **Setting** New fields added.

Date	AA Ver	Doc Ver	Change Details
			Rules (Admin UI) Rule Management section replaces previous ***Authentication Exemption*** a ***Registration*** sections Tabs for: Registration previously ***Force Registration*** tab under **Rules** Authentication previously ***Authentication Exemption*** tab under **Rules** Settings.
			Cards (Admin UI) **Users** tab renamed to **Cards**.
			Reports (Admin UI) Reports support reporting by 3-D Secure version.
			Transactions (Admin UI) **Find 3-D Secure**: supports search by 3-D Secure version. New fields adde
			Admins (Admin UI) Admin User Details and User Profile: added **2-factor authentication** login
			Local Messaging (Specifications) Changed Final Registration Request with OOB device registration request
			Remote Messaging (Specifications) Added **issuerName** and **theeDSProtocolVersion** in **Transaction** table Added **HeaderParams** table Added **AdditionalParams** table Added **AuthType** in **PreAuthResp** table Added new OTP types for **AuthType** and **oobInfo** in **InitAuthReq** table Sample Request Response: changed **CVD** to NULL.

Date	AA Ver	Doc Ver	Change Details
			CHANGES TO DOCUMENTATION STRUCTURE All documentation moved online with the ability to print to PDF ***To print the entire ActiveAccess documentation***: click the button on the Introduction page. ***To print a section***: click the button on that section. ***Tip***: hovering your mouse over the button will let you see which section w printed. **See Documentation change details** for full details of the changes in the documentation moving from PDF to online format.
26/02/2019	**7.4.6**	**7.4.6.1**	**Remote Messaging** Added **AuthType** in **initAuthReq** table Changed RegToken definition in **CardInfo** table.
06/07/2018	**7.4.0**	**7.4.0:1**	Addition of options in **System Management > Settings** to allow administ specified access levels to view Card Number (plaintext) and AAV/CAVV/AEV Changed description of Soft Launch List Addition of ActiveAccess Error Codes in Appendix A.

Documentation change details

Online Main Menu	Sub Menus	Previous PDF Document / Latest Changes
Introduction		
Installation Guide >		A11-Install_Maint_TechRef.pdf
	Product Architecture	
	External Components	
	Installation	
Administration UI >		AA12-ActiveAccess Administration.pdf
	About the Issuer Administration Server	AA12 / Added support for two-factor authentication for logging into the Administration UI
	System Management >	AA12
	About System Management	AA12
	Settings	AA12
	ACS Settings	AA12
	Issuer Management	AA12
	- Group Management	AA12
	- Authentication Mgmt >	**New Subsection**
	- About Authentication Management	**New**

Online Main Menu	Sub Menus	Previous PDF Document / Latest Changes
	- Devices	AA12, previously Device Management
	- Risk	**New**
	- OOB	**New**
	Public & Encryption Key Management	AA12
	Exchange Configuration	AA12
	Archive Management	AA12
	Security	AA12
	- Issuer Certificate	AA12
	- AHS Certificate	AA12
	- CAAS Certificate	AA12
	- Directory Server Certificate	**New**
	- OOB Certificate	**New**
	- Risk Certificate	**New**
	- CA Certificate	AA12
	Servers	AA12
	- MIA Servers	AA12
	- Access Control Servers (ACS)	AA12
	- Authentication History Servers (AHS)	AA12

Online Main Menu	Sub Menus	Previous PDF Document / Latest Changes
	- Centralised Authentication and Authorisation Servers (CAAS	AA12
	- Out of Band Authentication Servers (OOB)	AA12
	- Risk Servers	AA12
	Utilities >	
	Utilities	AA12
	Key Retiring Utility	AA12
	Issuers	AA12
	- Settings	AA12
	- Upload Registration Files	AA12
	- Custom Pages	AA12
	- Key Management	AA12
	Rules	
	- Registration -- Amount Threshold -- Merchant Blacklist	AA12
	- Authentication -- Soft Launch List Rule -- Merchant Whitelist Rule -- Merchant Watchlist -- Location Watchlist - Location Watchlist Search Results -- Domestic & International Transaction Amount Threshold -- Stand-In Transaction Threshold	AA12

Online Main Menu	Sub Menus	Previous PDF Document / Latest Changes
	- Settings	AA12
	Admin Users	AA12
	Cards	AA12 **Users** renamed to **Cards**
	Transactions	AA12
	Reporting	AA12
	Audit Log	AA12
	Profile Management_	AA12
Specifications		
	Local Messaging >	
	Local Messaging	AA61-Messaging Specification.pdf
	Card Loader	AA32-GPayments Card Loader.pdf
	Remote Messaging >	
	Remote Messaging	AA71-Remote System Messaging Specification.pdf
	Country and Currency Codes	AA71-Remote System Messaging Specification.pdf Appendix A
	Sample Card	AA71-Remote System Messaging Specification.pdf Appendix B
	Sample Request Response	AA71-Remote System Messaging Specification.pdf Appendix C
	SMS via JMS	AA83-ActiveAccess - SMS via JMS Library.pdf
	Out of Band Authentication Adapter	**New**

Online Main Menu	Sub Menus	Previous PDF Document / Latest Changes
	Risk Engine Adapter	**New**
Error Codes		AA12 - Appendix A
Glossary		AA12
Document Control>		
	Document Control	AA12
	Documentation Changes (*this page*)	**New**
Release Notes		Previously included in the ActiveAccess package
Legal Notices		AA12

Release Notes

ActiveAccess v9.0.4

[06/09/2021]

[EOL: Two years after the subsequent version's release date]

Type	Issue Number	External ID	Description	Components
ENHANCEMENT	#248		New Visa Secure Program Guide	Setup, Issuer Administration, Access Control Server, Registration Server, Whitelisting Server
ENHANCEMENT	#456		Visa PSD2 Exemptions - EMV 3DS Supplementary Guide	Access Control Server
ENHANCEMENT	#606		Visa Secure: RBA Adapter for Acquirer Country Code Extension	Access Control Server
ENHANCEMENT	#620		EMV 2.2: Support message Extension Device Acknowledgement	Access Control Server
ENHANCEMENT	#686		Visa Secure – Validation on Token ID&V Requests	Access Control Server
ENHANCEMENT	#687		Visa 3DS 2.X – Support for Travel Extension	Access Control Server
ENHANCEMENT	#774		Visa Secure: Release Updates - August 2021	Access Control Server
ENHANCEMENT	#806	#9413	Allowing 3DS1 messages with a sequence of elements that are not in order (optional)	Access Control Server
ENHANCEMENT	#829		CAVV for CEMEA Region	Access Control Server

Type	Issue Number	External ID	Description	Components
FIX	#807	#9456	Log file for OOB doesn't build on ACS	Access Control Server
FIX	#826	#9778	Merchant Activity report bug	Issuer Administration
FIX	#828	#9777	OTP input field doesn't appear on authentication pages	Access Control Server
FIX	#830	#9768	Purchase Date check 9.0.4 temporary fix	Access Control Server

ActiveAccess v9.0.3

[16/08/2021]

[EOL: 31/08/2023]

Type	Issue Number	External ID	Description	Components
ENHANCEMENT	#817		Support for importing CAVV and HMAC keys via MIA	Issuer Administration
FIX	#822	#9720	MIA and Registration server ping error	Issuer Administration, Registration Server

ActiveAccess v9.0.2

[06/08/2021]

[EOL: 13/08/2023]

Type	Issue Number	External ID	Description	Components
ENHANCEMENT	#818	#9720	Added additional HSM key logs	Issuer Administration

ActiveAccess v9.0.1

[04/08/2021]

[EOL: 06/08/2023]

Type	Issue Number	External ID	Description	Components
FIX	#814	#9720	Fixed CryptokiError: 0x40 encrypted data invalid error	Issuer Administration, Access Control Server, Registration Server, Whitelisting Server

ActiveAccess v9.0.0

[20/07/2021]

[EOL: 04/08/2023]

Type	Issue Number	External ID	Description	Components
ENHANCEMENT	#235		EMV 3DS 2.2: Whitelisting	Whitelisting Server
ENHANCEMENT	#317		Support for EMV 3D Secure 2.2	Issuer Administration, Access Control Server, Registration Server, Whitelisting Server
ENHANCEMENT	#381		EMV 3DS 2.2: Support for Decoupled Authentication, ARes.TransStatus=D	Access Control Server
ENHANCEMENT	#385		EMV 3DS 2.2: Support HTTP Protocol HTTP/1.1 and higher	Access Control Server
ENHANCEMENT	#386		EMV 3DS 2.2: Support Informational Request	Access Control Server

Type	Issue Number	External ID	Description	Components
ENHANCEMENT	#442	#9320	Update ACS UI Data Elements	Access Control Server
ENHANCEMENT	#458		Deprecated features: ActiveDevice/User Authentication, Enrolment Server, Device types: CAP and RSA	Issuer Administration, Access Control Server, Registration Server, CardLoader
ENHANCEMENT	#466		Mastercard Extension for RBA	Issuer Administration, Access Control Server
ENHANCEMENT	#473	#8958	EMV 3DS 2.2: Remove Continue button from OOB page - Local Issuer	Access Control Server
ENHANCEMENT	#474		EMV2.2: OOB authentication page content	Access Control Server
ENHANCEMENT	#501		EMV 3DS 2.2: Update shared key generation	Access Control Server
ENHANCEMENT	#507		Visa Secure: Support Authentication for Non-Payment Authentication	Access Control Server
ENHANCEMENT	#511	#9093	Implement general error pages	Issuer Administration
ENHANCEMENT	#526		Add option to configure cardholder name validation	Issuer Administration, Access Control Server, Registration Server
ENHANCEMENT	#573		Allow admin to enter adapterId when registering adapter	Issuer Administration

Type	Issue Number	External ID	Description	Components
ENHANCEMENT	#577		Visa secure: using CAVV algorithm U3V7 update status	Issuer Administration, Access Control Server
ENHANCEMENT	#579	#9161	MasterCard: Acquirer Strong Consumer Authentication (SCA) Exemption support in ACS	Access Control Server
ENHANCEMENT	#590		Visa Secure: Support for 3RI payments	Access Control Server
ENHANCEMENT	#592		Add option 3D Secure 2.2 in transaction search	Issuer Administration
ENHANCEMENT	#593		EMV 3DS 2.2: Reporting	Issuer Administration
ENHANCEMENT	#594		EMV2.2: Archive for new protocol version	Issuer Administration
ENHANCEMENT	#600		Visa Secure: Secure Corporate Payment (SCP)	Access Control Server
ENHANCEMENT	#607	#9022	SessionID logging	Access Control Server
ENHANCEMENT	#614		New Key Management and HSM connectivity - Phase II - Including Migrate to Data Key Utility	Issuer Administration, Access Control Server, Registration Server
ENHANCEMENT	#634		EMV 3DS 2.2: ThreeRI handling	Access Control Server
ENHANCEMENT	#684		Visa Secure: transStatusReason=21 no longer supported in VISA	Access Control Server

Type	Issue Number	External ID	Description	Components
ENHANCEMENT	#653		Selecting CAVV/IAV algorithm in Issuer Groups	Issuer Administration
ENHANCEMENT	#662		EMV 3DS 2.2: Whitelisting API	Issuer Administration, Access Control Server, Registration Server
ENHANCEMENT	#669	#9043	Create New CAVV/AAV/SPA Key as Inactive	Issuer Administration
ENHANCEMENT	#680	#9320	EMV 3DS 2.2: UI elements in CRes for SDK transactions	Access Control Server
ENHANCEMENT	#681		Support for Tomcat 9	Setup, Issuer Administration, Access Control Server, Registration Server, Whitelisting Server
ENHANCEMENT	#688	#9320	Support both portrait and landscape UI templates - Local Issuer	Access Control Server
ENHANCEMENT	#689	#9299	Dynamic Linking - SMS/Email OTP verification issue	Access Control Server
ENHANCEMENT	#696		Whitelisting API for audit logs	Whitelisting Server
ENHANCEMENT	#726	#9320	Support both portrait and landscape UI templates - Remote Issuer	Access Control Server
ENHANCEMENT	#798		EMV 3DS 2.2: EMVCo ReferenceNumber update	Setup
FIX	#337	#8044	Card registration flow - "Unable to assign device as it is not active"	Registration Server

Type	Issue Number	External ID	Description	Components
FIX	#487	#9035	Errors after successful completion of 3DS2 transaction	Access Control Server
FIX	#524	#9108	Custom pages do not scale	Access Control Server
FIX	#615	#9208	Incorrect purchase date processing by ACS	Access Control Server
FIX	#616	#9184	Invalid procedure FIND_NESTED_VISA_CAVV_FORMAT	Setup
FIX	#627	#9207	Authentication report bug: MIA Reports - Incorrect 3DS2 authentication data	Issuer Administration
FIX	#637		3DS2 archived transaction details	Issuer Administration
FIX	#730		Authentication with two SMS devices	Access Control Server
FIX	#731		Authenticate card with damaged/lost/ temporary disabled device	Access Control Server
FIX	#747	#9424	threeDSCompInd processing issue	Access Control Server
FIX			General fixes, performance and security enhancements	Setup, Issuer Administration, Access Control Server, Registration Server

ActiveAccess v8.5.10

[17/06/2021]

[EOL: 30/07/2023]

Type	Issue Number	External ID	Description	Components
FIX	#761	#9451	Error in MQ data processing for transactions where merchant name contains non-Latin letters	Access Control Server

ActiveAccess v8.5.9

[28/05/2021]

[EOL: 17/06/2023]

Type	Issue Number	External ID	Description	Components
FIX	#732	#9382	Data element not in the required format or value is invalid as defined	Access Control Server

ActiveAccess v8.5.8

[17/05/2021]

[EOL: 28/05/2023]

Type	Issue Number	External ID	Description	Components
ENHANCEMENT	#597	#9153	Change Display Amount Format for VND currency	Setup, Access Control Server
FIX	#676	#9313	Incorrect displaying sms parameters	Issuer Administration
FIX	#714	#9366	ACS: session expired	Access Control Server

ActiveAccess v8.5.7

[04/05/2021]

[EOL: 17/05/2023]

Type	Issue Number	External ID	Description	Components
FIX	#712	#9285	Issue in setting ClientId on queued SMS tokens	Access Control Server, Issuer Administration

ActiveAccess v8.5.6

[30/04/2021]

[EOL: 04/05/2023]

Type	Issue Number	External ID	Description	Components
ENHANCEMENT	#682	#9285	ClientID parameter in SMS via JMS	Access Control Server
FIX	#608	#9346	Element 'param' validation error	Access Control Server
FIX	#694	#9331	Successful authentication without CReq field in POST request	Access Control Server

ActiveAccess v8.5.5

[09/04/2021]

[EOL: 30/04/2023]

Type	Issue Number	External ID	Description	Components
ENHANCEMENT	#670	#9291	Visa Secure: Support Visa CEMEA Region CAVV generation for different transStatus values	Access Control Server
ENHANCEMENT	#678		EMV2.x: avoid padding in base64Url	Access Control Server

Type	Issue Number	External ID	Description	Components
FIX	#672	#9291	Error generating CAVV value in 3DS1 transaction	Access Control Server
FIX	#673	#9298	Local 3DS1 authentication issue with OOB devices	Access Control Server

ActiveAccess v8.5.4

[02/03/2021]

[EOL: 09/04/2023]

Type	Issue Number	External ID	Description	Components
FIX	#641	Errors during functional test of the interaction between the OOB Server and ACS	Access Control Server	
FIX	#656	ACS start up issue with the new OpenJDK Vendor Name	Setup	
FIX		General fixes, performance and security enhancements	Setup, Issuer Administration, Access Control Server, Registration Server	

ActiveAccess v8.5.3

[05/02/2021]

[EOL: 02/03/2023]

Type	Issue Number	External ID	Description	Components
ENHANCEMENT	#645	Making thread pool size configurable	Access Control Server	

Type	Issue Number	External ID	Description	Components
FIX	#626	Notification Report for current date	Registration Server	
FIX	#629	App-based authentication issue	Access Control Server	
FIX	#630	Incorrect value of $PurchaseDateTime in SMS messages	Access Control Server	
FIX	#632	EMV 3DS2.1 - Recurring transactions processing	Access Control Server	

ActiveAccess v8.5.2

[15/01/2021]

[EOL: 05/02/2023]

Type	Issue Number	Description	Components
FIX	#610	Fixed an Issue in creating certificate for AnyBank during setup	Access Control Server
FIX		General fixes, performance and security enhancements	Setup, Issuer Administration, Access Control Server, Registration Server

ActiveAccess v8.5.1

[24/12/2020]

[EOL: 15/01/2023]

Type	Issue Number	Description	Components
FIX		General fixes, performance and security enhancements	Setup, Issuer Administration, Access Control Server, Registration Server

ActiveAccess v8.5.0

[18/12/2020]

[EOL: 24/12/2022]

Type	Issue Number	Description	Components
ENHANCEMENT	#422	Enabling migration of ACS application server from Tomcat to WebLogic	Issuer Administration, Access Control Server, Registration Server, Enrolment Server
ENHANCEMENT	#463	New Key Management and HSM connectivity - Phase I	Setup, Issuer Administration, Access Control Server, Registration Server, Enrolment Server
ENHANCEMENT	#468	Support for Oracle 19c	Setup, Issuer Administration, Access Control Server, Registration Server, Enrolment Server
ENHANCEMENT	#522	Addition of purchaseDate to CAAS Server's oobInfo	Access Control Server, Issuer Administration
ENHANCEMENT	#543	Mask critical data in log	Access Control Server
ENHANCEMENT	#557	Improved RMI support	Issuer Administration
FIX	#372	Incorrect CRes transStatus when RReq communication failed	Access Control Server
FIX	#431	New issuer creation error	Setup, Issuer Administration, Access Control Server
FIX	#445	CAVV U3v0 for RBA EMV 3DS	Access Control Server, Issuer Administration

Type	Issue Number	Description	Components
FIX	#459	SMS counter issue when card has multiple devices	Access Control Server
FIX	#494	Extended logs for xslTransform not finished	Access Control Server
FIX	#555	Ending OOB transaction when not authenticated	Access Control Server
FIX	#556	threeDSReqAuthData missing	Access Control Server
FIX	#561	CAAS 3DS2 back issue	Access Control Server
FIX	#567	Set label `Challenge` for C&R in 3DS2 pages	Access Control Server
FIX	#583	Invalid date and time in authentication landing page (2.1 version)	Access Control Server
FIX	#596	CardLoader/Registration API: can't load cards	Registration Server
FIX	#598	billAddrState, shipAddrState field validation (ISO 3166-2 codes)	Access Control Server
FIX	#599	SMS Templates	Issuer Administration
FIX	#601	OOB without continue button - Shutdown issue	Access Control Server
FIX	#602	ACS should display OOB Continue button when WS is unreachable	Access Control Server
FIX	#603	OOB without continue button - reduce CLOSE_WAIT time	Access Control Server
FIX		General fixes, performance and security enhancements	Setup, Issuer Administration, Access Control Server, Registration Server

ActiveAccess v8.4.4

[25/11/2020]

[EOL: 18/12/2022]

Type	Issue Number	Description	Components
ENHANCEMENT	#550	Update Risk Engine Integrated in CAAS	Access Control Server
FIX	#572	SDK issue for remote issuer	Access Control Server
FIX	#574	HMAC256 key creation error for Luna Provider	Access Control Server, Issuer Administration

ActiveAccess v8.4.3

[13/11/2020]

[EOL: 25/11/2022]

Type	Issue Number	Description	Components
FIX	560	Fixed 3DS1 remote authentication issue when authType = 10	Access Control Server
FIX	563	Fixed issue of formatting purchase date in CAAS API logs	Access Control Server
FIX	564	Fixed acs.war issue of formatting purchase date displaying in Remote/Local issuer authentication challenge page	Access Control Server

ActiveAccess v8.4.2

[27/10/2020]

[EOL: 13/11/2022]

Type	Issue Number	Description	Components
ENHANCEMENT		Enhancement on the remote issuer custom pages: both 3DS1 and 3DS2 remote authentication custom pages should be uploaded	Access Control Server
FIX	549	Added version in schema.xsd at acs.war/WEB-INF/lib/caas.client-*.jar	Access Control Server
FIX	552	Restore authType compatibility: authType can be used for authentication methods 1-15	Access Control Server

ActiveAccess v8.4.1

[16/10/2020]

[EOL: 27/10/2022]

Type	Issue Number	Description	Components
ENHANCEMENT	#528	Support multi-instance for OOB Notifier	Access Control Server
FIX	#485	Update authentication methods	Access Control Server
FIX	#514	Mastercard 3DS2.1: generation of authentication method dropdown on the page	Access Control Server
FIX	#525	PAReq - invalid session	Access Control Server
FIX	#530	Issue with adding sms-centers to issuers on MIA	Issuer Administration
FIX	#533	Issue retrieving the wsUrl (Remote Issuer)	Access Control Server
FIX		General fixes, performance and security enhancements	Setup, Issuer Administration, Access Control Server, Registration Server

ActiveAccess v8.4.0

[02/10/2020]

[EOL: 16/10/2022]

Type	Issue Number	Description	Components
ENHANCEMENT	#348	Support new Visa Secure CAVV Usage 3, Version 7 and add an option to select the algorithm	Access Control Server, Issuer Administration
ENHANCEMENT	#383	Separate SDK html pages from BRW html pages	Access Control Server
ENHANCEMENT	#387	Remove Continue button & add support for auto-submission of OOB page - Remote Authentication	Access Control Server, Issuer Administration
ENHANCEMENT	#455	Display IAV generation algorithm in Transaction Details	Access Control Server, Issuer Administration
ENHANCEMENT	#457	Extend OOB Adapter Challenge Request API with purchase date and time element	Access Control Server
ENHANCEMENT	#467	Assign multiple SMS devices to cards that have different SMSC	Registration Server
ENHANCEMENT	#482	Configurable log for number of DB connections	Access Control Server
ENHANCEMENT	#498	Compatibility with Visa authentication page requirements	Access Control Server
FIX	#437	Pages do not stretch to the entire height of the device - AnyBank_Remote Custompages_3DS2 - incorrect page display	Access Control Server
FIX	#440	Error during decryption in CardDeviceUpdate	CardLoader, Registration Server

Type	Issue Number	Description	Components
FIX	#454	Failed 3DS2 transaction details in MIA	Access Control Server, Issuer Administration
FIX	#460	Error during retrieving messageExtension from session	Access Control Server
FIX	#462	Actions for when OobAuthenticationResult indicates cardholder did not perform OOB auth or there was a connection issue	Access Control Server
FIX	#483	SessionID logging	Access Control Server
FIX	#486	CAVV issue - PAN length must be 16	Access Control Server
FIX	#492	Amount without separator	Access Control Server
FIX	#493	Fix $PurchaseDateTime format in SMS messages	Access Control Server
FIX	#494	The xslTransform not finished	Access Control Server
FIX	#495	3DS2 Challenge errors flow	Access Control Server
FIX	#499	Incorrect data in MIA Reports	Issuer Administration
FIX	#503	Error during parsing sessionInfo when cardId is UUID for Remote Issuers	Access Control Server
FIX	#504	Remote page issue - OOB initAuth error	Access Control Server
FIX	#509	SDK sessionKey should be saved in DB	Access Control Server
FIX		General fixes, performance and security enhancements	Setup, Issuer Administration, Access Control Server, Registration Server

ActiveAccess v8.3.6

[07/08/2022]

[EOL: 02/10/2022]

Type	Issue Number	Description	Components
ENHANCEMENT	#450	Save valid messages with Invalid ISO codes	Access Control Server
FIX	#437	Text displayed incorrectly when token is entered on Remote Authentication pages	Access Control Server
FIX	#446	Display issue for 3DS1 Local Authentication when OOB + SMS was assigned to the card	Access Control Server
FIX	#447	Disabled the validation of cardholder name for 3DS2 authentication	Access Control Server
FIX		General fixes, performance and security enhancements	Setup, Issuer Administration, Access Control Server, Registration Server

ActiveAccess v8.3.5 (Patch)

[16/07/2020]

[EOL: 07/08/2022]

Type	Issue Number	Description	Components
FIX	#441	AAV generation issue for 3DS1 Mastercard transactions	Access Control Server

ActiveAccess v8.3.4 (Patch)

[09/07/2020]

[EOL: 16/07/2022]

Type	Issue Number	Description	Components
FIX	#441	Removing cancel button in XSL pages for SDK transactions	Access Control Server

ActiveAccess v8.3.3 (Patch)

[06/07/2020]

[EOL: 09/07/2022]

Type	Issue Number	Description	Components
ENHANCEMENT	#441	Additional logs added for 3DS1 Mastercard transactions	Access Control Server

ActiveAccess v8.3.2 (Patch)

[26/06/2020]

[EOL: 06/07/2022]

Type	Issue Number	Description	Components
FIX	#441	Extending the Message Length for SDK transactions	Access Control Server

ActiveAccess v8.3.1 (Patch)

[12/06/2020]

[EOL: 26/06/2022]

Type	Issue Number	Description	Components
ENHANCEMENT	#441	Additional logs added for SDK transactions	Access Control Server

ActiveAccess v8.3.0

[29/05/2020]

[EOL: 12/06/2022]

Type	Issue Number	Description	Components
ENHANCEMENT	#158	OTP & password option for OOB	Issuer Administration, Access Control Server
ENHANCEMENT	#274	Encrypting critical data such as cardnumber in adapters	Issuer Administration, Access Control Server
ENHANCEMENT	#325	Encryption of card number in CardLoader logs	CardLoader
ENHANCEMENT	#343	IAV method option for Mastercard PSD2 in Issuer groups	Issuer Administration, Access Control Server
ENHANCEMENT	#328	RMI configuration option	Setup, Issuer Administration, Access Control Server
ENHANCEMENT	#403	Add 3DS2 transactional data into CAAS messages	Access Control Server
ENHANCEMENT	#423	MIA to notify user when device is removed from Issuer's Active Device list	Issuer Administration
ENHANCEMENT	#429	Remove case sensitivity of OobRequestChallengeResult.requestChallengeEnum accepted values	Access Control Server
FIX	#370	OOB deviceId length issue	Registration Server

Type	Issue Number	Description	Components
FIX	#373	FileNotFoundException during RBA and OOB startup	Access Control Server
FIX	#421	10-CR challenge authentication issue	Access Control Server
FIX	#427	Updated ECI values for AMEX, JCB and Diners	Access Control Server
FIX	#438	Change SecureCode HMAC 256 key	Issuer Administration
FIX		General fixes, performance and security enhancements	Setup, Issuer Administration, Access Control Server, Registration Server

ActiveAccess v8.2.6 (Patch)

[07/05/2020]

[EOL: 29/05/2022]

Type	Issue Number	Description	Components
FIX	#375	Stop ACS from uploading CustomPages for AnyBank at start up	Issuer Administration
FIX	#420	NullPointer Exception during SMS device loading	Access Control Server
FIX	#426	MIA Report error	Setup, Issuer Administration, Access Control Server, Registration Server

ActiveAccess v8.2.5 (Patch)

[04/05/2020]

[07/05/2022]

Type	Issue Number	Description	Components
FIX	#420	NullPointer Exception during SMS device loading	Access Control Server

ActiveAccess v8.2.4 (Patch)

[28/04/2020]

[04/05/2022]

Type	Issue Number	Description	Components
FIX	#420	NullPointer Exception during SMS device loading	Access Control Server

ActiveAccess v8.2.3 (Patch)

[24/04/2020]

[28/04/2022]

Type	Issue Number	Description	Components
FIX	#419	Issue with ACS authentication pages and authentication results cannot be seen	Access Control Server

ActiveAccess v8.2.2

[17/04/2020]

[24/04/2022]

Type	Issue Number	Description	Components
FIX	#371	Fixes to Frictionless Flow, Browser, PA (Result = N)	Access Control Server
FIX	#412	Luna HSM KeyStore loading issue	Access Control Server, Setup
FIX	#413	RSA key size for new issuers and issuer groups changed to 2048	Access Control Server, Setup
FIX	#416	Fixes to Frictionless Flow, 3RI, and NPA (Result = Y)	Access Control Server

ActiveAccess v8.2.1

[09/04/2020]

[EOL: 17/04/2022]

Type	Issue Number	Description	Components
ENHANCEMENT	#331	Addition of cancel link to 3DS2 authentication pages	Access Control Server
ENHANCEMENT	#369	Addition of "store name", "date" and "amount" to authentication page	Access Control Server
FIX	#349	null cardName in verifyRegResp produces an error	Access Control Server
FIX	#371	Changes to the validation date of cardLoader generated certificate	CardLoader
FIX	#393	Misplacement of elements in responsive view of custom pages	Access Control Server
FIX	#394	NullPointerException error while processing regStatus=1 in CAAS	Access Control Server

Type	Issue Number	Description	Components
FIX	#395, 400	ClientID=null not to be included in Notification Reports, OOB & RBA APIs	Access Control Server, CardLoader, Registration Server
FIX	#396	Exception during initializing LunaProvider in gpcomp.updater	Setup
FIX	#397, #399	Archive database schema upgrade from ActiveAccess v7.3 to ActiveAccess v8.2	Issuer Administration, Setup
FIX	#407	Configuration of "ACS challenge URL" for issuers	Access Control Server

ActiveAccess v8.2.0

[27/03/2020]

[EOL: 09/04/2022]

Type	Issue Number	Description	Components
ENHANCEMENT	#151	Support push notifications during OOB authentication	Access Control Server, Registration Server
ENHANCEMENT	#174	IAV method option for Mastercard PSD2	Access Control Server, Issuer Administration
ENHANCEMENT	#192	Displaying OTP+StaticPassword for CAAS	Access Control Server
ENHANCEMENT	#221	Displaying risk decision in Transaction Details page	Issuer Administration
ENHANCEMENT	#307	Addition of a new card attribute: ClientID	Access Control Server, CardLoader, Issuer Administration, Registration Server

Type	Issue Number	Description	Components
ENHANCEMENT	#316	Card and Transaction search performance improvement	Issuer Administration
ENHANCEMENT	#319	"Score range for device" in RBA allows for selection from all devices including OOB	Access Control Server
ENHANCEMENT	#323	Addition of "Maximum interaction" limit for Remote Issuers	Access Control Server, Issuer Administration
FIX	#234	Fix for CAASSESSION table lock issue	Access Control Server
FIX	#315	Fix for archive and purge features	Issuer Administration
FIX	#353	Reverting Card Expiry Date to optional	Issuer Administration, Registration Server
FIX		General fixes, performance and security enhancements	Setup, Issuer Administration, Access Control Server, Registration Server

ActiveAccess v8.1.2

[10/01/2020]

[EOL: 28/02/2022]

Type	Issue Number	Description	Components
ENHANCEMENT	#228	Adding forgot password link for browser device channel	Access Control Server
ENHANCEMENT	#251	Send tokens only when the Resend OTP link is clicked	Access Control Server
ENHANCEMENT	#268	Changes to PreAuth in Remote Authentication model	Access Control Server

Type	Issue Number	Description	Components
ENHANCEMENT	#299	Improvements to enabling 2FA for admin users	Issuer Administration
ENHANCEMENT	#300	Device selection when two OOB devices are assigned to a card	Access Control Server
ENHANCEMENT	#312	Addition of DESede support to CardLoader and Registration for backward compatibility	Registration Server, CardLoader
FIX	#271	Fixing Ping Command connection issue	Issuer Administration, Access Control Server, Registration Server, Enrolment Server
FIX		General fixes, performance and security enhancements	Setup, Issuer Administration, Access Control Server, Registration Server

ActiveAccess v8.1.1

[06/12/2019]

[EOL: 10/01/2022]

Type	Issue Number	Description	Components
ENHANCEMENT	#267	Add new CancelReg request with optional cardholder name	Registration Server, CardLoader
ENHANCEMENT	#271	ActiveAccess Ping command improvement	Issuer Administration, Access Control Server, Registration Server, Enrolment Server
FIX	#303	Invalidate empty cardholder name in PreReg and FinalReg	Registration Server, CardLoader

ActiveAccess v8.1.0

[15/11/2019]

[EOL: 06/12/2021]

Type	Issue Number	Description	Components
ENHANCEMENT	#92	Acceptable values for `App unsupported devices` updated	Access Control Server, Issuer Administration
ENHANCEMENT	#131	Supporting two-factor authentication for local authentication	Access Control Server, Issuer Administration
ENHANCEMENT	#142	Changing the risk/rule decision process	Access Control Server
ENHANCEMENT	#143	Provide a mechanism to test OOB and RBA restful adapters connect/read timeouts	Access Control Server, Issuer Administration
ENHANCEMENT	#179	Including more data in RBA call back	Access Control Server
ENHANCEMENT	#198	Updating the approach of populating the historical transaction for RBA	Access Control Server
ENHANCEMENT	#201	Create a swagger for OOB and Risk restful adapters	Access Control Server
ENHANCEMENT	#246	Enabling language selection during authentication for 3DS1	Access Control Server, Issuer Administration
ENHANCEMENT	#273	Http protocol version for external connections	Access Control Server
FIX	#53	3DS method notification post data	Access Control Server
FIX	#95	ACS decision based on risk chain score in remote authentication	Access Control Server
FIX	#260	HSM installation issues	Setup

Type	Issue Number	Description	Components
FIX	#266	Detach SDK certificates from Issuer Certificates	Setup
FIX	#278	CAAS Server throws NullPointer when message category is NPA	Access Control Server

ActiveAccess v8.0.4

[06/11/2019]

[EOL: 15/11/2021]

Type	Issue Number	Description	Components
FIX	#281	Invalid Request to Remote Server	Access Control Server

ActiveAccess v8.0.3

[25/10/2019]

[EOL: 06/11/2021]

Type	Issue Number	Description	Components
FIX	#277	Deployment of registration.war during startup	Registration
FIX	#278	CAAS throws a NullPointer when message category is NPA	Access Control Server

ActiveAccess v8.0.2

[09/10/2019]

[EOL: 25/10/2021]

Type	Issue Number	Description	Components
ENHANCEMENT	#51	Support 3DS2 purchase amount 0 for Mastercard IDC	Access Control Server
ENHANCEMENT	#98	Update ECI for Message Category NPA for Mastercard IDC	Access Control Server
ENHANCEMENT	#219	Making acsReferenceNumber configurable for testing purposes	Issuer Administration, Access Control Server
ENHANCEMENT	#223	Addition of decline code to preAuthResp of CAAS	Access Control Server
ENHANCEMENT	#229	Addition of KeyStore and TrustStore for RBA Server	Access Control Server
ENHANCEMENT	#233	Addition of KeyStore and TrustStore for OOB Server	Access Control Server
FIX	#132	Updates to Mastercard IDC status codes	Access Control Server
FIX	#148	Remote CAAS PreAuth changes	Access Control Server
FIX	#226	Setup could not generate RSA2048 keys for the MAP error during Luna PKCS11 installation/upgrade	Setup
FIX	#242	Verified by Visa references changed to Visa Secure in the content of authentication pages	Access Control Server
FIX		General fixes, performance and security enhancements	Setup, Issuer Administration, Access Control Server, Registration Server

ActiveAccess v8.0.1

[05/09/2019]

[EOL: 02/10/2021]

Type	Issue Number	Description	Components
ENHANCEMENT	#169	EULA update	Issuer Administration
ENHANCEMENT	#208	Grant scripts run automatically during setup	Setup
FIX	#172	Device selection page isn't being shown	Access Control Server
FIX	#182	Device registration fails when issuer has OOB device enabled	Access Control Server
FIX	#186	Exception raised during Diners Club remote authentication	Access Control Server
FIX	#188	ChallengeResponse failure in remote authentication	Access Control Server
FIX	#189	Risk adapter configuration page issue	Issuer Administration
FIX	#193	Generate RSA 2048 when the EC key generation fails	Setup, Issuer Administration, Access Control Server
FIX	#196	CardLoader setup.sh doesn't work	CardLoader
FIX	#203	Upgrade issue from 7.4.2 to 8.0.0 with currency exchange rate	Setup
FIX		General fixes, performance and security enhancements	Setup, Issuer Administration, Access Control Server, Registration Server

ActiveAccess v8.0.0

[15/08/2019]

[EOL: 05/09/2021]

Type	Issue Number	Description	Components
ENHANCEMENT	#93	Enhancements to the Administration interface (MIA)	Issuer Administration
ENHANCEMENT	#5468	Support incremental database schema changes in Setup	Setup
ENHANCEMENT	#5801	Web Container Neutralization	Setup
ENHANCEMENT	#6659	Support for 3-D Secure 2.1	Setup, Issuer Administration, Access Control Server, Registration Server
ENHANCEMENT	#6661	3DS2 Transaction search based on 3DS version	Issuer Administration
ENHANCEMENT	#6663	Support for 3DS2 Risk Management	Issuer Administration, Access Control Server
ENHANCEMENT	#6664	Support 3DS2 Reporting	Issuer Administration
ENHANCEMENT	#7207	Support for OOB Processing	Issuer Administration, Access Control Server
ENHANCEMENT	#7383	Substitute Triple DES encryption in ActiveAccess with stronger cryptography	Issuer Administration, Access Control Server
ENHANCEMENT	#7845	Removal of RuPay component	Setup, Issuer Administration
ENHANCEMENT	#7880	Two-factor authentication for MIA login	Issuer Administration
ENHANCEMENT	#8082	Simplify the setup process	Setup
ENHANCEMENT	#8310	SPA2 algorithm for AAV generation	Setup, Issuer Administration, Access Control Server
FIX	#5425	MIA allows exceeded password length and updates it successfully	Access Control Server

Type	Issue Number	Description	Components
FIX	#7297	Adminlog and AuditlogCollectorErrors have been updated to fix the errors that occurred during scheduler job	Access Control Server
FIX	#8160	Authentication Exemption Rules for CAAS server	Access Control Server

ActiveAccess v7.4.7 (Patch)

[23/03/2019]

[EOL: 15/08/2021]

Access Control Server		
FIX	#8147	Fixed the purchAmount field to avoid the mismatch of value between PARes and PAReq

ActiveAccess v7.4.6 (Patch)

[05/03/2019]

[EOL: 23/03/2021]

Issuer Administration		
FIX	#8022	Removing "+" sign when sending message via JMS.

Access Control Server		
FIX	#8022	Removing "+" sign when sending message via JMS.

ActiveAccess v7.4.5 (Patch)

[01/02/2019]

[EOL: 05/03/2021]

Access Control Server		
ENHANCEMENT	#7843	Displaying the Mobile Number on Remote Authentication pages.
ENHANCEMENT	#7893	Adding PurchaseExponent attribute to the transaction table of requests to CAAS.

ActiveAccess v7.4.4 (Patch)

[27/09/2018]

[EOL: 01/02/2021]

Issuer Administration		
FIX	#7748	SMS delivery fails as ACS sends the phone number without the '+' sign to SMPP client. ACS now includes the + sign when sending SMS.

Access Control Server		
FIX	#7748	SMS delivery fails as ACS sends the phone number without the '+' sign to SMPP client. ACS now includes the + sign when sending SMS.

ActiveAccess v7.4.3 (Patch)

[18/09/2018]

[EOL: 27/09/2020]

Issuer Administration		
FIX	#7718	Card Registration File Upload Errorcard file. Clearing the timer to prevent "java.lang.IllegalStateException: Timer already canceled" exceptions.

ActiveAccess v7.4.2

[20/08/2018]

[EOL: 07/06/2020]

Issuer Administration		
ENHANCEMENT	#7543	ISO 3166 Update country details for Eswatini
ENHANCEMENT	#7654	ISO 4217 Amendment Number 169

Active Control Server		
ENHANCEMENT	#7543	ISO 3166 Update country details for Eswatini
ENHANCEMENT	#7654	ISO 4217 Amendment Number 169
FIX	#7677	CurrencyExchange error in ActiveAccess startup

Registration Server		
FIX	#7639	Card Registration File Upload

ActiveAccess v7.4.1 (Patch)

[08/08/2018]

[EOL: 20/08/2020]

Issuer Administration		
FIX	#7557	Verification code not received for Email device type

Active Control Server		
FIX	#7482	Custom Pages layout updates

Active Control Server		
FIX	#7557	Verification code not received for Email device type

ActiveAccess v7.4.0

[06/07/2018]

[EOL: 08/08/2020]

Setup		
ENHANCEMENT	#6479	External HSM setup - PKCS #11 Support
ENHANCEMENT	#7470	Update key type for CVC2 process
ENHANCEMENT	#7471	HMAC key length update for MC
ENHANCEMENT	#7477	Support HSMs in which DES is not available
ENHANCEMENT	#7519	Upgraded log4j from 1.2.13 to the 1.2.17 version
FIX	#7380	Visa 3-D Secure Security Program - Encryption of CAVV/AAV values
FIX	#7518	Updated GET_CARDS procedure

Issuer Administration		
ENHANCEMENT	#6479	External HSM setup - PKCS #11 Support
ENHANCEMENT	#7359	ISO 4217 Amendment Number 166
ENHANCEMENT	#7470	Update key type for CVC2 process
ENHANCEMENT	#7471	HMAC key length update for MC
ENHANCEMENT	#7477	Support HSMs in which DES is not available
ENHANCEMENT	#7519	Upgraded log4j from 1.2.13 to the 1.2.17 version

Issuer Administration		
FIX	#7329	Public key for the Issuer Group
FIX	#7380	Visa 3-D Secure Security Program - Encryption of CAVV/AAV values
FIX	#7520	Purge processor is already running error

Access Control Server		
ENHANCEMENT	#6479	External HSM setup - PKCS #11 Support
ENHANCEMENT	#7359	ISO 4217 Amendment Number 166
ENHANCEMENT	#7482	Combining two device registration custom pages into one
ENHANCEMENT	#7519	Upgraded log4j from 1.2.13 to the 1.2.17 version
FIX	#7047	Updating the path of caaswarning.properties to keep it unchanged during the upgrade process
FIX	#7380	Visa 3-D Secure Security Program - Encryption of CAVV/AAV values
FIX	#7518	Updated GET_CARDS procedure

Enrolment Server		
ENHANCEMENT	#6479	External HSM setup - PKCS #11 Support
ENHANCEMENT	#7519	Upgraded log4j from 1.2.13 to the 1.2.17 version

Registration Server		
ENHANCEMENT	#6479	External HSM setup - PKCS #11 Support
ENHANCEMENT	#7519	Upgraded log4j from 1.2.13 to the 1.2.17 version

ActiveAccess v7.3.3 (Patch)

[25/05/2018]

[EOL: 06/07/2018]

Access Control Server		
FIX	#7402	Incorrect JCB transaction status with 'Card Not Found' from CAAS

ActiveAccess v7.3.2 (Patch)

[29/03/2018]

[EOL: 25/05/2020]

Access Control Server		
FIX	#7160	Remove error on missing MD field

ActiveAccess v7.3.1 (Patch)

[20/02/2018]

[EOL: 29/03/2020]

Access Control Server		
FIX	#7116	JCB VEReq with Browser.deviceCategory=1

ActiveAccess v7.3.0

[29/01/2018]

[EOL: 20/02/2020]

Setup		
FIX	#6334	Correction to the casing for SafeNet in setup/sample.ini
FIX	#6338	Remove WebSphere application server option from setup
FIX	#6986	Decryption error during notification report process
FIX	#7052	Notification reports - java.lang.NullPointerException

Issuer Administration		
FIX	#6406	Exception thrown when clicking Back on Matched Rule Details page
FIX	#6244	Update the default value for AMEX 'Maximum forgot password attempts
FIX	#6620	MIA incorrectly searches the WEB-INF folder for cacerts, instead of the config folder
FIX	#6645	Cards do not get assigned to the most detailed BIN
FIX	#7052	Notification reports - java.lang.NullPointerException
ENHANCEMENT	#4131	Authentication pages compatibility with mobile devices
ENHANCEMENT	#5935	New authentication method Email OTP
ENHANCEMENT	#6252	ISO 3166 Update country details for Moldova and Gambia
ENHANCEMENT	#6308	Addition of a message on MIA's blank screen for admin users of Issuers with an invalid license key
ENHANCEMENT	#6377	Option to defer application of Setting changes to next server restart
ENHANCEMENT	#6463	ISO 4217 Currency Code Service - Amendment number 163
ENHANCEMENT	#6527	Mastercard Identity Check Support
ENHANCEMENT	#6688	JCB Attempt process

Issuer Administration		
ENHANCEMENT	#6727	Security enhancements
ENHANCEMENT	#6765	All PANs must now comply with the Luhn algorithm and pass a Mod-10 check
ENHANCEMENT	#6773	ISO 4217 Amendment Number 164
ENHANCEMENT	#6823	Rules Settings challenge option for 'not exempted authentications' as per IDC requirements
ENHANCEMENT	#6981	ISO 4217 Amendment Number 165

Access Control Server		
FIX	#5686	Proof of Attempt = Disabled still displays the opt-out link during ADS
FIX	#6244	Update the default value for AMEX 'Maximum forgot password attempts
FIX	#6417	PAReq is not logged by ACS when the Authentication Exemption Rules are used
FIX	#6687	Updating error details wording to match 3DS v1.0.2 document
FIX	#6693	Errors related to JCB compliance test
FIX	#7037	Authentication Exemption rules do not apply during transactions
ENHANCEMENT	#4131	Authentication pages compatibility with mobile devices
ENHANCEMENT	#5935	New authentication method Email OTP
ENHANCEMENT	#6209	Style applied to XML formatted error pages displayed during authentication
ENHANCEMENT	#6252	ISO 3166 Update country details for Moldova and Gambia
ENHANCEMENT	#6463	ISO 4217 Currency Code Service - Amendment number 163
ENHANCEMENT	#6527	Mastercard Identity Check Support

Access Control Server		
ENHANCEMENT	#6652	Compliance with JCB J/Secure
ENHANCEMENT	#6688	JCB Attempt process
ENHANCEMENT	#6689	Addition of new data elements in JCB Authentication page and updates to the masking format of PAN
ENHANCEMENT	#6691	Remove AHS support for JCB
ENHANCEMENT	#6692	Multi-language support of JCB pages
ENHANCEMENT	#6727	Security enhancements
ENHANCEMENT	#6765	All PANs must now comply with the Luhn algorithm and pass a Mod-10 check
ENHANCEMENT	#6773	ISO 4217 Amendment Number 164
ENHANCEMENT	#6823	Rules Settings challenge option for 'not exempted authentications' as per IDC requirements
ENHANCEMENT	#6981	ISO 4217 Amendment Number 165

Enrolment Server		
ENHANCEMENT	#6705	The effect of 'Uses confirmation' field in Enrolment
ENHANCEMENT	#6727	Security enhancements

Registration Server		
FIX	#6396	CardLoader error message does not correspond with Registration logs
ENHANCEMENT	#5935	New authentication method Email OTP
ENHANCEMENT	#6527	Mastercard Identity Check Support

Registration Server		
ENHANCEMENT	#6727	Security enhancements

ActiveAccess v7.2.1

[20/04/2017]

[EOL: 29/01/2020]

Setup v7.2.1

Issuer Administration v7.2.1

Access Control Server v7.2.1

Enrolment Server v7.2.1

Registration Server v7.2.1

Setup		
ENHANCEMENT	#6289	Encode hsmpassword parameter (Base64) in RuPay config file.

Issuer Administration		
FIX	#4584	PCI Key Retiring utility performance issue.
FIX	#6182	Certificate creation failure.
ENHANCEMENT	#6289	Encode hsmpassword parameter (Base64) in RuPay config file.

Access Control Server		
FIX	#4584	PCI Key Retiring utility performance issue.
FIX	#6186	Error while processing a custom page.
ENHANCEMENT	#4217	Addition of JCB XSL pages into the standard release package.

Access Control Server		
ENHANCEMENT	#6289	Encode hsmpassword parameter (Base64) in RuPay config file.

Enrolment Server		
ENHANCEMENT	#6289	Encode hsmpassword parameter (Base64) in RuPay config file.

Registration Server		
ENHANCEMENT	#6289	Encode hsmpassword parameter (Base64) in RuPay config file.

ActiveAccess v7.2.0

[22/12/2016]

[EOL: 20/04/2019]

Setup v7.2.0

Issuer Administration v7.2.0

Access Control Server v7.2.0

Enrolment Server v7.2.0

Registration Server v7.2.0

Rupay v1.1.0

Card Loader 1.1.41

Setup		
SUPPORT:	#5806	nCipherKM.jar being removed in installation
ENHANCEMENT:	#5474	Support silent mode installation
ENHANCEMENT:	#5939	Encode HSM_PASSWORD parameter (Base64) in ActiveAccess config files

Setup		
ENHANCEMENT:	#5574	Remove usage of deprecated JRE classes
FEATURE:	#5546	Supports Amex Safekey compliance (rev 2016)

Issuer Administration		
FIX:	#5525	Encrypt critical data in case of registration failure
FIX:	#5899	Archive history details page display error
SUPPORT:	#5729	Visa Intermediate SHA2 CA cert added for new installations
ENHANCEMENT:	#5574	Remove usage of deprecated JRE classes
ENHANCEMENT:	#5740	Exclusion of third party XML parser libraries (JAXP libraries),Third party XML parser libraries (JAXP libraries) excluded and replaced with JDK JAXP libraries
ENHANCEMENT:	#5829	Remove restriction on using previous CAVV key
ENHANCEMENT:	#5874	Support p7 and der files when installing certificates
ENHANCEMENT:	#5939	Encode HSM_PASSWORD parameter (Base64) in ActiveAccess config files
FEATURE:	#5546	Supports Amex Safekey compliance (rev 2016)

Access Control Server		
FIX:	#4584	Improve PCI Key Retiring utility performance*
FIX:	#5965	CAAS Card Auth Data format not found error. The error message is logged in ACS logs during a remote transaction regardless of success of the transaction.
FIX:		Various spelling corrections in application and XSL files

Access Control Server		
SUPPORT:	#5748	Error in restarting Number of authentication exemptions and Sum of exempted authentications' amounts when empty cardholder name is received from CAAS server
SUPPORT:	#5785	Unable to establish connection to CAAS
SUPPORT:	#5903	Optimise GET_CARDS procedure
SUPPORT:	#5952	Update American Express SafeKey logo
ENHANCEMENT:	#5054	Support SafeNet Network HSM (Cloud HSM/Luna SA)
ENHANCEMENT:	#5546	Compliance with American Express Safekey (revision 2016)
ENHANCEMENT:	#5574	Remove usage of deprecated JRE classes
ENHANCEMENT:	#5740	Exclusion of third party XML parser libraries (JAXP libraries),Third party XML parser libraries (JAXP libraries) excluded and replaced with JDK JAXP libraries
ENHANCEMENT:	#5939	Encode HSM_PASSWORD parameter (Base64) in ActiveAccess config files
FEATURE:	#5546	Supports Amex Safekey compliance (rev 2016)

Enrolment Server		
FIX:		Various spelling corrections in application and XSL files
ENHANCEMENT:	#5574	Remove usage of deprecated JRE classes
ENHANCEMENT:	#5740	Exclusion of third party XML parser libraries (JAXP libraries),Third party XML parser libraries (JAXP libraries) excluded and replaced with JDK JAXP libraries
ENHANCEMENT:	#5939	Encode HSM_PASSWORD parameter (Base64) in ActiveAccess config files

Registration Server		
SUPPORT:	#5767	Changing request Id length in notification request to be at most 1024 characters
ENHANCEMENT:	#5574	Remove usage of deprecated JRE classes
ENHANCEMENT:	#5740	Exclusion of third party XML parser libraries (JAXP libraries),Third party XML parser libraries (JAXP libraries) excluded and replaced with JDK JAXP libraries
ENHANCEMENT:	#5939	Encode HSM_PASSWORD parameter (Base64) in ActiveAccess config files

RuPay		
FIX:	#5482	Search by Error Code field in Transaction screens
FIX:	#6025	RuPay verifyRegistration did not forward contextBlob to initAuthentication. contextBlob now included
FIX:	#6026	Support authType in addition to authTypeSupList in RuPay

Card Loader		
FIX:	#5779	CardLoader now supports Java 8
SUPPORT:	#5767	Changing request Id length in notification request to be at most 1024 characters
ENHANCEMENT:	#5574	Remove usage of deprecated JRE classes

ActiveAccess v7.1.4

[03/10/2016]

[EOL: 22/12/2018]

Setup v7.1.4

Issuer Administration v7.1.4

Access Control Server v7.1.4

Enrolment Server v7.1.4

Registration Server v7.1.4

Issuer Administration		
Support	#5703	Database connectivity issue
Bug	#5720	ActiveAccess 7.1.4 beta 5 installation error: no record found
Enhancement	#5715	Version class in ActiveAccess should be filtered in Maven
Support	#5664	Login issue with remote issuers' business and helpdesk admins without access to rules
Support	#5548	FileNotFoundException: auditconfig.properties changed from an Error to a Warning
Bug	#5745	CSR Export Issue

Access Control Server		
Support	#5703	Database connectivity issue
Bug	#5689	CAAS: ISO currency & country codes
Enhancement	#5523	Risk Based Authentication
Bug	#5674	DB Warning Logger in ACS log file
Enhancement	#5715	Version class in ActiveAccess should be filtered in Maven
Enhancement	#5688	Copyright of XSL pages
Bug	#5685	AHS logging PATransReq twice in the acs log file
Support	#5646	Merchant URL Must be URL pattern

Access Control Server		
Support	#5634	PARes with parameter SSID to MPI
Support	#5616	A null priSec value results in NullPointerException
Enhancement	#5596	Support for unmasked CH.fullPAN in PATRANSReq messages

Enrolment Server		
Enhancement	#5715	Version class in ActiveAccess should be filtered in Maven

Registration Server		
Enhancement	#5715	Version class in ActiveAccess should be filtered in Maven

Setup		
Bug	#5735	RuPay tables missing in database after installation
Enhancement	#5715	Version class in ActiveAccess should be filtered in Maven
Bug	#5678	RuPay module being installed without being selected (Centos 6.x)
Bug	#5562	No rupay WAR files found in tomcat/webapps when installing AA with Rupay option

ActiveAccess v7.1.3

[03/09/2016]

[EOL: 03/10/2018]

Setup v7.1.3

Issuer Administration v7.1.3

Access Control Server v7.1.3

Enrolment Server v7.1.3

Registration Server v7.1.3

Access Control Server		
Bug	#5619	SignatureMethod must be SHA1

No changes in other components

Legal Notices

Confidentiality Statement

GPayments reserves all rights to the confidential information and intellectual property contained in this document. This document may contain information relating to the business, commercial, financial or technical activities of GPayments. This information is intended for the sole use of the recipient, as the disclosure of this information to a third party would expose GPayments to considerable disadvantage. No part of this document may be reproduced, stored in a retrieval system or transmitted in any form or by any process without prior written permission. This information is provided under an existing non-disclosure agreement with the recipient.

Copyright Statement

All third party product and service names and logos used in this document are trade names, service marks, trademarks, or registered trademarks of their respective owners.

The example companies, organizations, products, people and events used in screenshots in this document are fictitious. No association with any real company, organization, product, person, or event is intended or should be inferred.

Disclaimer

GPayments Pty Ltd makes no, and does not intend to make any, representations regarding any of the products, protocols or standards contained in this document. GPayments Pty Ltd does not guarantee the content, completeness, accuracy or suitability of this information for any purpose. The information is provided "as is" without express or implied warranty and is subject to change without notice. GPayments Pty Ltd disclaims all warranties with regard to this information, including all implied warranties of merchantability and fitness for a particular purpose and any warranty against infringement. Any determinations and/or statements made by GPayments Pty

Ltd with respect to any products, protocols or standards contained in this document are not to be relied upon.

Liability

In no event shall GPayments Pty Ltd be liable for any special, incidental, indirect or consequential damages whatsoever (including, without limitation, damages for loss of business profits, business interruption, loss of business information, or any other pecuniary loss) whether in an action of contract, negligence or other tortuous action, rising out of or in connection with the use or inability to use this information or the products, protocols or standards described herein, even if GPayments has been advised of the possibilities of such damages.

GPayments