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
  - Authentication Server
  - Verify Enrolment Server
  - Challenge Server
  - RMI Server
  - AHS Client
  - Rules Engine
  - External Messaging Adapter
  - Risk Engine Adapter
  - Out of Band Authentication Adapter
- Administration Server
- Registration Server
- Enrolment 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 **3-D Secure** and **ActiveDevice protocols**.
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.
ActiveDevice is a device agnostic protocol for strong authentication of online users, which uses a variety of two-factor authentication techniques.

Authentication Server

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 authentication server is used for user authentication in 3-D Secure and ActiveDevice processes. The user is redirected to the authentication server by the merchant plug-in during the 3-D Secure process and by the ActiveDevice plug-in in the two-factor authentication. The authentication pages are stored in the database and served via the authentication server itself.

The authentication server is responsible for processing of the PAReq and generation of PARes message pair in the 3-D Secure process.

The authentication server is responsible for processing of the UAReq and generation of UARes message pair in the ActiveDevice process.

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, DPI (ActiveDevice Plug-In)

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 and ActiveDevice processes. The verify enrolment server consumes VEReq and UEReq messages and generates VERes and UERes 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.

Challenge Server (3DS2)

Default path: /acs/ca

Inbound connections: User's browser, 3DS SDK app

RMI Server

Default port: 4242 and 4241

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).

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 (Issuers 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 and user 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 users (pre-registration and final registration models).

Enrolment Server

A fully customisable enrolment website, which allows cardholders to enrol their cards with the authentication schemes.

Default port: Determined by the application server

Default path: /enrolment/

Protocol: HTTP/HTTPS

Inbound connections: User's browser

Outbound connections: Database server

Other requirements: Must be able to access the HSM

The enrolment pages are stored in the database. These pages are customised per issuer. The enrolment server uses XSL to combine issuer's customised look and feel and enrolment process with the cardholder enrolment and authentication criteria provided as XML.

The enrolment server is only used for enrolment of pre-registered cardholders with static password to allow them to participate in authenticated e-commerce transactions via 3-D Secure 1 protocol.

Database Server

Default port: 1521

Default path: N/A

Protocol: TCP

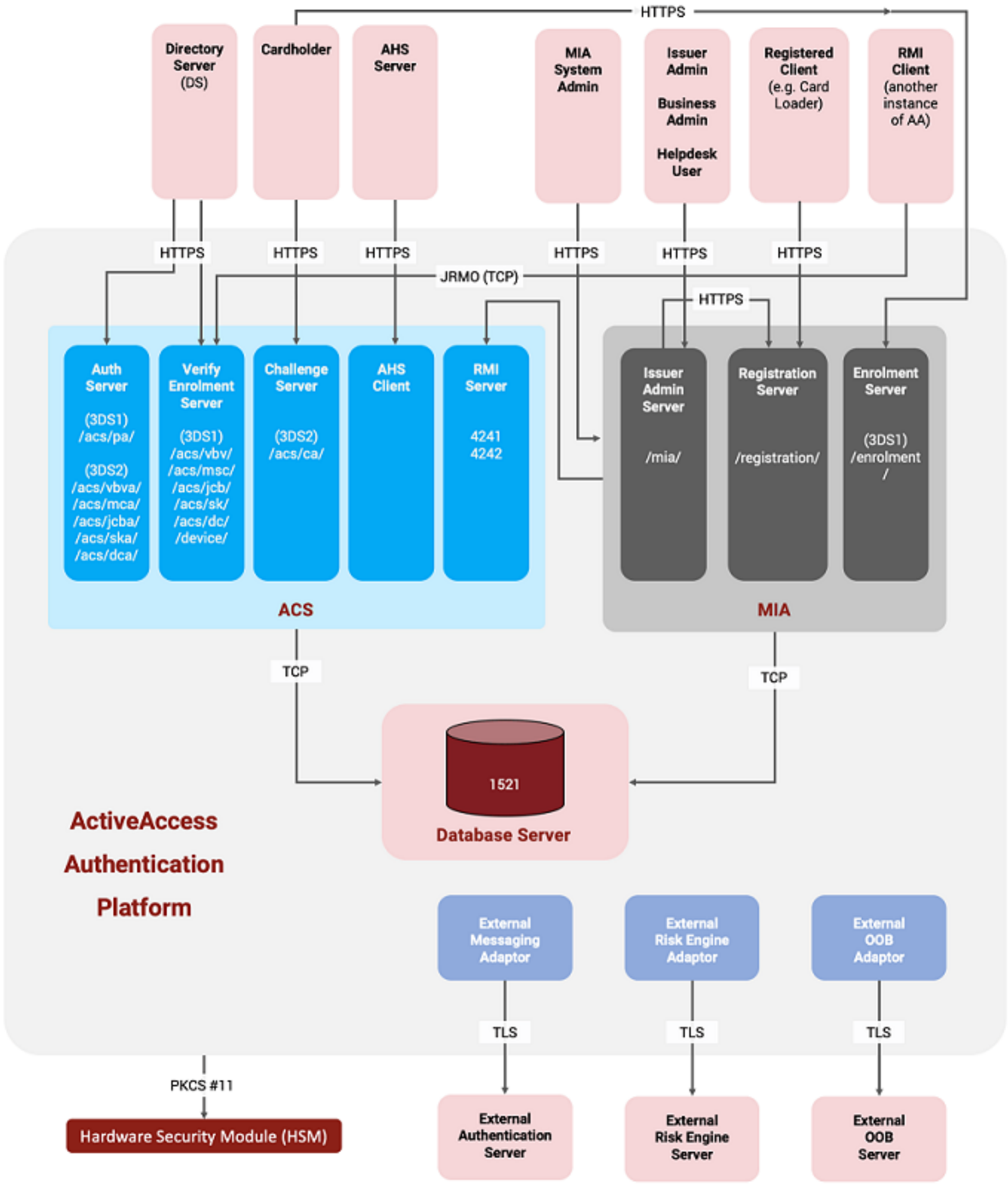
Inbound connections: Authentication server, Verify enrolment server, RMI Server, AHS Client, Rule Engine, External Messaging Adapter, Administration server, Registration server, Enrolment server.

Outbound connections: None

Other requirements: None

Logical View of ActiveAccess

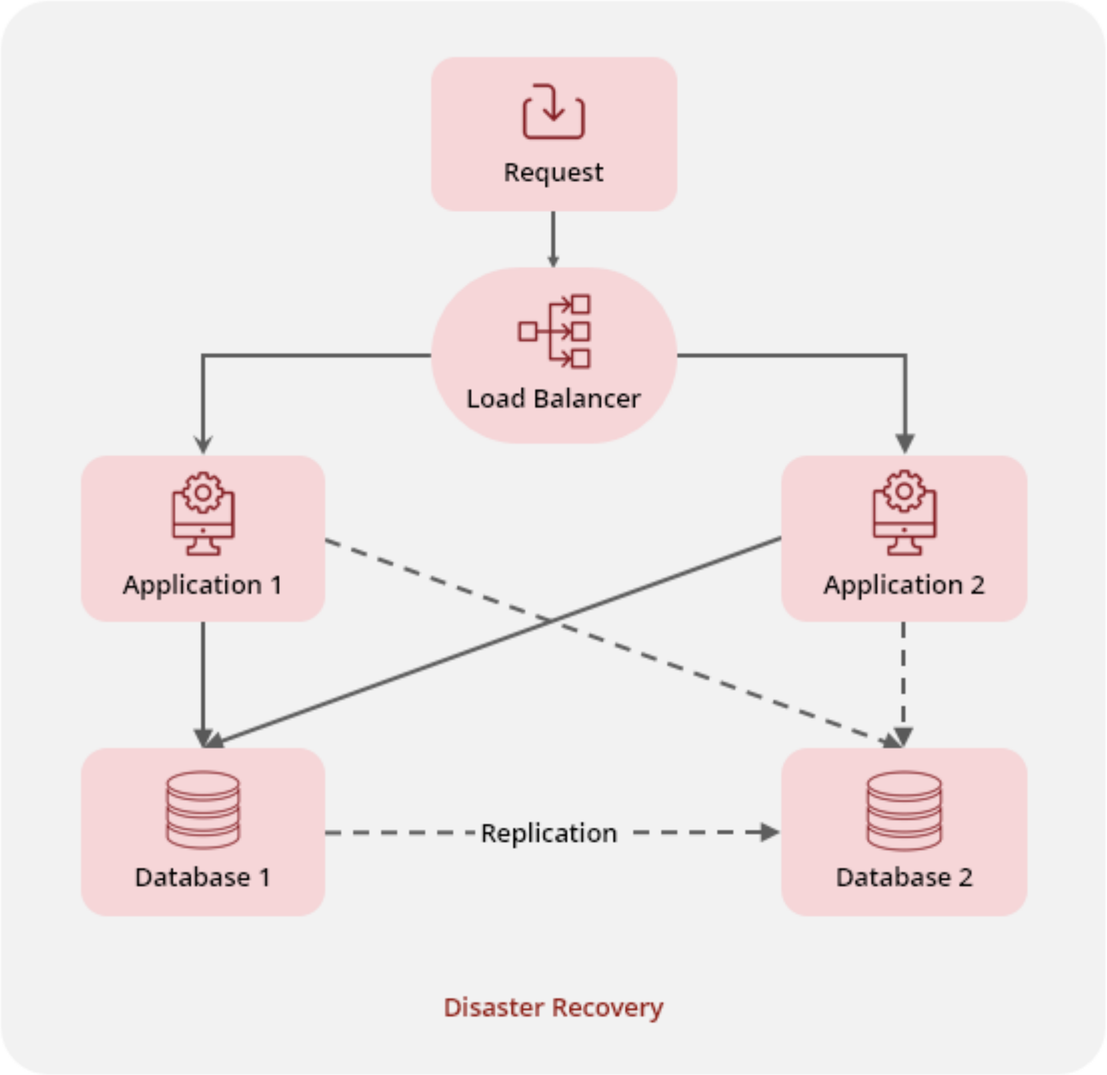
The following diagram displays the logical view of ActiveAccess with the components explained earlier on this page.



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.



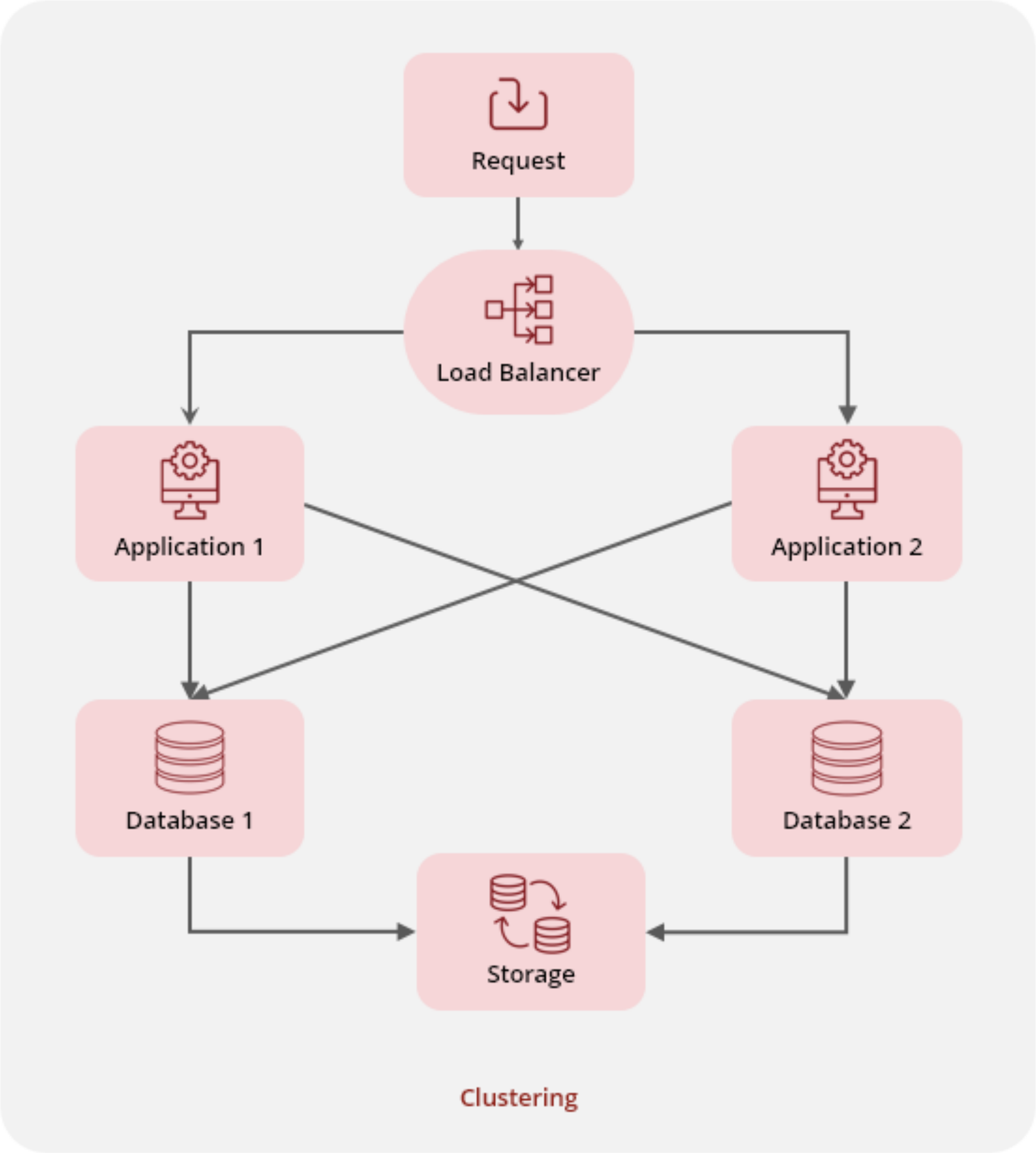
Production Setup with Clustering

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	- Java Application Servers compatible with Servlet specification 3.0 (e.g. Tomcat 7.0.x and later)
Database	- Oracle 11g - 11gXE - 12c

[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 and ActiveAccess+RuPay require 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" />
```

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 the 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
```

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 AUTHSESSION 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

CAP

Currently two CAP schemes are supported: M/Chip 4 and M/Chip 2.1. CAP functionalities are supported only with the *Thales e-Security* HSM device. The *Thales e-Security* HSM module must be setup to support EMV functionalities (nShield / SPP).

CAP KEYS

Appropriate CAP keys must be created for an issuer that requires CAP support. The keys must be manually created in the HSM using the key management facilities provided by the HSM vendor.

Issuer keys must follow particular naming conventions as follows:

- For M/Chip2.1: `cap2mchip< Issuer_ID >`
- For M/Chip 4: `cap4mchip< Issuer_ID >`

  where `< Issuer_Id >` specifies the Issuer ID of the corresponding issuer as assigned by ActiveAccess.

When creating the keys select key roles mkac2r and mkac4r for M/Chip 2.1 and M/Chip 4, respectively. You also need to specify a field named IIPB by SPP module which is the AC part of the CAP IPB (Issuer Proprietary Bitmap).

Please refer to 'Key-loading Solutions Guide' by *Thales e-Security*, for further information on creating and handling keys.

SOFTWARE MODE

For testing purposes only ActiveAccess can run CAP in software emulation mode, without the need for setting up CAP keys in the HSM. The CAP emulation mode is only available for M/Chip 4.

- In order to run ActiveAccess in CAP emulation mode, create a text file containing the CAP keys. The file may contain a key entry for each issuer in the form:

  ```
  <key_alias>=<key_value>
  ```

  where < key_value > is the value of key expressed in hexadecimal format. For example

  ```
  cap4mchip1234567890=9E15204313F7318ACB79B90BD986AD29
  ```

- Now save the file and give an arbitrary name. Assuming that the file is named 'capkeys.values' and stored in '/opt/activeaccess' directory, you need to all the following line to ActiveAccess start up script:

  -Dcom.gpayments.CAPKeys.file = /opt/activeaccess/capkeys.values

> **Note**
>
> Software mode is only provided for test purposes and must not be used in production.
>
> You cannot use CAP in hardware while software mode is enabled. Be sure to remove reference to your CAP key file, if you wish to use hardware for M/Chip 4 or M/Chip 2.1.

CAP LOGGING

CAP uses the global java logger to log the CAP related activities. So by setting the java.util.logging.config.file property to an arbitrary java logging configuration file, you can have different levels of logging (Severe, Warning, Info, Fine, Finer, Finest, All) for CAP authentications. More detail is output when ActiveAccess is run in CAP simulation mode.

RSA

To Enable RSA devices, you need to download and copy the RSA Java library file (RSASecurIDAuthenticationEngineAPI.jar) site to the library directory of ActiveAccess application server. You may need to contact RSA Security in order to receive the Java library file.

RSA token keys should be uploaded in the system. These files are provided by RSA and can be uploaded to ActiveAccess using the administration interface.

- Browse to **System Management** > **Device Management** choose **upload file** and then specify the file and relevant parameters.

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.

Installation

> *ActiveAccess installation and setup processed simplified*

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 aliases `AA_Administration`, `MIA_DB_DESede` and the issuer key aliases (e.g. `VbVA< Issuer_ID >`, `VbVB< Issuer_ID >`, `RSAVbV< Issuer_ID >`, `ECVbV< Issuer_ID >`, etc) have 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

Download and extract the ActiveAccess installation package, provided by GPayments.

Installation parameters

- 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.

Upgrades from v7.4.x

> **Before the upgrade:**
>
> 1. Shutdown all instances of ActiveAccess, stop the current Tomcat servers.
> 2. Back up ActiveAccess directories. Archive the ActiveAccess directory and store in a safe place. Do this for all instances of ActiveAccess.
> 3. Back up the Tomcat application server directories. Archive directories where the application has been deployed and store in a safe place.
> 4. 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.
> 5. Back up all the HSM key data.

- Add the following line in the **acsconfig.properties** file (located in **TOMCAT_HOME/bin/config**)

  `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`, `eb_config.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`.

- Δ Back up and remove the following files from **TOMCAT_HOME/lib**
  - gpcomp.pki-1.1.5-3.jar
  - gpcomp.hsm-1.2.24-0.jar
- + Copy the following files from the ActiveAccess installation package in **ActiveAccess/jars** to **TOMCAT_HOME/lib**
  - gpcomp.hsm-1.2.27-0.jar

    - gpcomp.pki-1.1.7-1.jar

- ➕ 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.

New installations

- In the installation package, go to the **ActiveAccess** directory, copy **activeaccess.properties** and paste it in your `AA_HOME` directory.
- Open **activeaccess.properties** and fill in the required configuration parameters.

> ⚠ **Warning**
>
> For upgrades, you should not copy the **activeaccess.properties** file in your **AA_HOME**. This file gets generated automatically.

Deploying WAR packages

Access Control Server, Administration Server, Enrolment Server and Registration Server are distributed as WAR packages. To install these packages, deploy **acs.war**, **enrolment.war**, **mia.war** and **registration.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 and pasted to **TOMCAT_HOME/webapps**. 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**.

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 Enrolment 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**
    - Enrolment Server: **enrolment.war**
- Configure the **installation parameters** (**AA_HOME** directory and configuration file).
- Start the application server.
- Ensure that the AES (128 Bits) key aliases `MIA_DB_DESede`, `AA_Administration`, `Card< Issuer_ID >` and the issuer key aliases (e.g. `VbVA< Issuer_ID >`, `VbVB< Issuer_ID >`,

`RSAVbV< Issuer_ID >`, `ECVbV< Issuer_ID >`, etc. for all card scheme providers) exist in the HSM.

> **Tip**
>
> It is important to note that the issuers and issuer keys are generated on the local HSM used by the Administration server.
>
> If you are installing a component on the same machine as the Administration server, these keys will be created. However, if these servers are installed on physically separate machines that use their own HSM, you will need to export these keys from the Administration server HSM and import them to the local HSM of the Access Control Server, Enrolment and Registration servers' HSM devices.
>
> **Do not attempt to create the keys directly as it will result in creation of physically different keys and the component will not be able to interact with the database server.**

Post Installation

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	/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
ActiveDevice authentication	/device	/pa		

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 blank page will be displayed. Otherwise, if the ACS is down, an error will be displayed.

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 blank page will be displayed. Otherwise, if the Administration Server is down, an error will be displayed.

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 blank page will be displayed. Otherwise, if the Registration Server is down, an error will be displayed.

Enrolment Server

Base URL: `https://< serveraddress >:< port >/enrolment/< IssuerID >`

Info

The Enrolment Server uses a unique URL for each issuer. When an issuer is created, it is assigned a unique, system generated Issuer ID. Enrolment pages can only be viewed after an issuer has been successfully enrolled and the enrolment package for that issuer has been uploaded to the system through the Administration server.

Monitoring the availability of Enrolment: `https://< serveraddress >:< port >/enrolment/ping`

Info

If the Enrolment Server is up and running, a blank page will be displayed. Otherwise, if the Enrolment Server is down, an error will be displayed.

Configuration Files

ActiveAccess Configuration File

/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.

> **Example**
>
> ```
> #   library=/opt/foo/lib/libpkcs11.so
> #   slot=1
> #   name=Safenet
> ```

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.

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.

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`.

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.

Additional Administration Server Configuration Parameters

UPLOADCACHE_DIR

Used to specify a location to copy uploaded file that VASCO and RSA 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 VACSO or RSA token files before an error is returned. In other words, if processing a VASCO or RSA 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.

MODULE

Used for initialising of the key manager for CAP functions. Select HSM for secure computation and cryptographic functions. A value of zero results in load sharing among all nShield capable modules. Default value is **0**.

PSINAME

Used for initializing the key manager for CAP functions. It is the name of the nShield installation to be initialized. Default value is **gpaymentsTest**.

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.

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**.

MODULE

Used for initialising of the key manager for CAP functions. Select HSM for secure computation and cryptographic functions. A value of zero results in load sharing among all nShield capable modules. Default value is **0**.

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

Additional Enrolment Server Configuration Parameters

CACHE:

Specify the caching mode used for caching issuer pages. (0: every visit, 1: automatically, 2: never, default value is **0**.)

MAX_CACHE:

Specify the number of issuer pages that will be cached. Default value is: 100.

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

providerName	providerIndex	cName	providerId
JCB	3	jcb	3
AMEX	4	sk	5
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.

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			
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
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

Server Error Codes			
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
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

Server Error Codes			
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
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

User Error Codes			
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
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

User Error Codes			
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
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

User Error Codes			
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
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

Account Error Messages		
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
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

Account Error Messages		
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
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

Authentication Device Messages		
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
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

Authentication Device Messages		
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
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

Authentication Device Messages		
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
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

Local Pages Errors	
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
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

Local Pages Errors	
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
303	Invalid one time password
304	Invalid serial number
305	Device is lost
306	Device is damaged
307	Device is already assigned

Local Pages Errors	
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
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

Local Pages Errors	
607	Device is damaged
608	Device is already assigned

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.

Document Control

+ new item Δ item changed X item removed 🟦 no change to item

Date	AA Ver	Doc Ver	Change Details
[05/09/2019]	8.0.1	8.0.1:1	**Product Architecture** (Installation Guide) Δ Disaster Recovery and Clustering diagrams added.
			Installation (Installation Guide) Δ Changes made to Upgrades from v7.4.x and New installations.
			Security (Admin UI) + Create Certificate Request: New **Key type** field added.
			Risk Engine Adapter (Specifications) Δ **ParameterDataElements**: **Validator** field description updated Δ **RemoteAssessmentRequest Data Elements**: **PreviousData** field format updated + **AReqWithTransStatusDataElements** added Δ **AReq Data Elements**: **ThreeDSCompInd** and **ThreeDSRequestorAuthenticationInd** field updated.
			Remote Messaging (Specifications) Δ **InitAuthReq** table: Usage of **oobInfo** changed.
			Out of Band (OOB) Authentication Adapter (Specifications) Δ Change the URL in **Restful API version of OOB Adapter** Δ Change `NOT__AUTHENTICATED` to `NOT_AUTHENTICATED` Δ Update MobilePhone Data Elements, HomePhone Data Elements, and WorkPhone Data Elements.

Date	AA Ver	Doc Ver	Change Details
15/08/2019	**8.0.0**	**8.0.0:1**	**Product Architecture** (Installation Guide) Δ Components labelled with (3DS1) or (3DS2) as relevant + Challenge Server (3DS2) added. + Risk Engine Adapter added + Out of Band (OOB) Authentication Adapter added Δ Logical view of ActiveAccess diagram updated Δ Hardware and Software Requirements updated X Removed references to **RuPay** components.
			External Components (Installation Guide) Δ Application Server dependency removed, supports compatible Java Application Servers.
			Installation (Installation Guide) Δ ActiveAccess installation and setup process simplified.
			System Management (Admin UI) + **Authentication Management** section added with tabs for: Δ Device Management previously under **System Management** + Risk Management for 3DS2 risk management + OOB Management for OOB processing support.
			System Management (Admin UI) - **Issuer Management** Δ Device Settings: OOB added as a supported device.
			Security (Admin UI) + Directory Server Certificate section added + OOB Certificate section added + Risk Certificate section added.
			Issuers (Admin UI) Δ Providers parameters moved to a new page, and linked, from the **Settings** page. New fields added.

Date	AA Ver	Doc Ver	Change Details
			Rules (Admin UI) Δ Rule Management section replaces previous ***Authentication Exemption*** and ***Force Registration*** sections Tabs for: + Registration previously ***Force Registration*** tab under **Rules** Δ Authentication previously ***Authentication Exemption*** tab under **Rules** 🟦 Settings.
			Cards (Admin UI) Δ **Users** tab renamed to **Cards**.
			Reports (Admin UI) Δ Reports support reporting by 3-D Secure version.
			Transactions (Admin UI) Δ **Find 3-D Secure**: supports search by 3-D Secure version. New fields added.
			Admins (Admin UI) Δ Admin User Details and User Profile: added **2-factor authentication** login option
			Local Messaging (Specifications) Δ Final Registration Request: updated with OOB device registration request.
			Remote Messaging (Specifications) Δ **Transaction** table: **issuerName** and **theeDSProtocolVersion** added + **HeaderParams** table added + **AdditionalParams** table added + **PreAuthResp** table: **AuthType** added + **InitAuthReq** table: new OTP types for **AuthType** and **oobInfo** added Δ 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 will be printed. Δ **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** Δ **initAuthReq** table: added AuthType Δ **CardInfo** table: RegToken definition updated.
06/07/2018	7.4.0	7.4.0:1	⊞ Addition of options in **System Management > Settings** to allow administrators at specified access levels to view Card Number (plaintext) and AAV/CAVV/AEVV Δ Updated 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	Δ 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 Δ **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		Δ Previously included in the ActiveAccess package
Legal Notices		AA12

Release Notes

ActiveAccess v8.0.1

[05/09/2019]

[EOL: Two years after the subsequent version's release date]

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

Type	Issue Number	Description	Components
FIX		Minor bug fixes, performance and security enhancements	Setup, Issuer Administration, Access Control Server, Registration Server

ActiveAccess v8.0.0

[15/08/2019]

[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

Type	Issue Number	Description	Components
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
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
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

Issuer Administration		
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	#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

Issuer Administration		
ENHANCEMENT	#6463	ISO 4217 Currency Code Service - Amendment number 163
ENHANCEMENT	#6527	Mastercard Identity Check Support
ENHANCEMENT	#6688	JCB Attempt process
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

Access Control Server		
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
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
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.
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
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*

Access Control Server		
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
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

Enrolment Server		
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

Access Control Server		
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
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

This work is Copyright © 2003-2019 by GPayments Pty Ltd. All Rights Reserved. No permission to reproduce or use GPayments Pty Ltd copyright material is to be implied by the availability of that material in this or any other document.

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.