

Sentinel CLOUD SERVICES™

Quick Start Guide



Software Version

This documentation is applicable for Sentinel Cloud 3.4.

Document Revision History

Part Number 007-012143-001

Revision F

Document Revision	Change	Date
A	Sentinel Cloud version 3.0 Release	December 2012
B	Sentinel Cloud 3.1 release. New features include: <ul style="list-style-type: none">▪ Server authentication with Public CA certificate and client authentication with message signing.	April 2013
C	Sentinel Cloud 3.2 release.	June 2013
D	Sentinel Cloud 3.3 release.	September 2013
E	Sentinel Cloud 3.3 release with on-premise support on Linux	October 2013
F	Sentinel Cloud 3.4 release	December 2013

Disclaimer and Copyrights

Copyright © 2013, SafeNet, Inc. All rights reserved.

<http://www.safenet-inc.com/>

We have attempted to make these documents complete, accurate, and useful, but we cannot guarantee them to be perfect. When we discover errors or omissions, or they are brought to our attention, we endeavor to correct them in succeeding releases of the product.

SafeNet® and Sentinel® are registered trademarks of SafeNet, Inc. All other product names referenced herein are trademarks or registered trademarks of their respective manufacturers.

Confidential Information

The Sentinel Cloud Services system is designed to protect your applications from unauthorized use. The less information that unauthorized people have regarding your security system, the greater your protection. It is in your best interest to protect the information herein from access by un-authorized individuals.

Table of Contents

Chapter 1: Quick Start	1
Steps to Get Started	1
What's Included in the E-mail Messages?	2
Downloading and Installing SDK	4
Sentinel Cloud Run-time Samples	5
Chapter 2: C Reference Application	7
Introduction	7
Files Included in the Sample	8
Windows	8
Linux	8
Prerequisites for Using Sample	9
Supported Operating Systems and Compilers	9
Required Libraries	9
Required Client Configuration File	9
Linux Specific Requirements	10
Required EMS Entities	10
Running the C Sample	11
Windows	11
Linux	12
Using the Run-time C APIs	14
Chapter 3: Java Demo Application	15
Introduction	15
Installing and Configuring Java Demo Application	16
System Requirements	16
Installation and Configuration	16
Accessing the Java Demo Application	19
Before You Use the Java Demo Application...	20
Creating Entitlements in the Sentinel EMS	20
Creating Users for the Cloud Demo Application	20

Using the Java Demo Application Products	21
Viewing Usage Related Information	21
Using the Sentinel Cloud Run-time Java APIs	22
Error Messages	23
Building the Cloud Java Demo Application	24
Using Eclipse	24
Using the Ant Build Script	25
Frequently Asked Questions - Java Demo Application	26
Question: How to uninstall the Cloud Demo Application?	26
Question: How to change the admin default password?	26
Chapter 4: .NET Reference Application	27
Introduction	27
Location of .NET Reference Application	28
Software Requirements	29
Before You Use the .NET Reference Application...	30
Building the .NET Reference Application	31
Accessing the .NET Reference Application	34
Using the .NET Reference Application	35
Viewing Usage Related Information	35
Using the Sentinel Cloud Run-time .NET APIs	36
Commonly Used Terms	39
Obtaining Support	45

Quick Start

Sentinel® Cloud Services is a robust licensing and entitlements management solution for your applications. It helps in feature-based user provisioning, authorization, metering, and management of your software delivered over cloud or on-premise.

This guide contains step by step instructions on how to quickly start using Sentinel® Cloud Services.



Steps to Get Started

1. [Get familiar with the items received in SafeNet e-mails](#)
2. [Download and install SDK](#)
3. [Explore the Sentinel Cloud Run-time samples](#)

What's Included in the E-mail Messages?

You receive two e-mails from SafeNet on purchase of Sentinel Cloud Services. The first e-mail contains SDK download link, EMS credentials, and Secret Key. The other one has the YPS Link and other configuration related information.

Here is a consolidated list of the items you receive in e-mails from SafeNet:

#	Items in the E-mails	Description
1	SDK Download Link	<p>This is the link from where you can download your Sentinel Cloud Services SDK - for C/.NET/Java platform.</p> <div style="border: 1px solid black; padding: 5px;">  This link remains available for a limited duration. </div>
2	Entitlement Management System (EMS) URL and Credentials	<p>When you subscribe for Sentinel Cloud Services, an EMS Server instance is configured for you. Use the URL and credentials given in e-mail to access EMS. (Type the URL in your Web browser, enter username and password, and click Submit.)</p> <div style="border: 1px solid black; padding: 5px;">  You need EMS for various entitlement management functions, like creating product catalogs, defining license strategies, provisioning contracts, etc. For details, you can refer to <i>Sentinel Cloud EMS User's Guide</i>, available in the <i>Manuals</i> folder of SDK. </div>
3	Secret Key	<p>The Secret Key is used for authenticating client requests. The Secret Key is included as a zipped attachment, which is protected with a password. The Secret Key zip file and password of zip file are sent in separate e-mails for security reasons.</p> <p>You will need to specify this value in the SecretKey property of Cloud Run-time configuration file.</p>
4	Secret Key ID	<p>The unique identifier of Secret Key.</p> <p>You will need to specify this value in the SecretKeyId property of the Cloud Run-time configuration file.</p>
5	Sentinel Directory Services URL (YPS Address)	<p>Address of Sentinel Directory Services, which is the server used for registration and authentication of Sentinel Cloud components.</p> <p>You will need to specify this address in the YPSAddress property of Cloud Run-time configuration file.</p>
6	Alias	<p>It is the vendor name provided in the vendor registration request file.</p> <p>You will need to specify this value in the ClientAlias property of the Cloud Run-time configuration file.</p>
7	Vendor ID	<p>The unique ID assigned to a software vendor.</p> <p>You will need to specify this value in the VendorId property of the Cloud Run-time configuration file.</p>

Please contact SafeNet Customer Support if anything is missing.

Notes:



- You will need items 3 to 7 while doing configuration settings for Sentinel Cloud Run-time APIs and Sentinel Cloud Samples.
- For details on Sentinel Cloud components mentioned in the table above (such as, EMS, Sentinel Cloud Directory Services, and Sentinel Cloud Run-time), refer to the [Sentinel Cloud Run-time Guide](#). This guide is also available in the *Manuals* folder of SDK.

Downloading and Installing SDK

The link to download Sentinel Cloud Services SDK is sent to you in an e-mail. Please see Section ["What's Included in the E-mail Messages"](#), for details about the items included in e-mail messages.

The downloaded SDK contains *Installation Guide*. Please refer it for installation details, such as:

- Media contents
- System requirements
- Installation steps
- Installed components

Sentinel Cloud Run-time Samples

After installing Run-time, you can explore various samples available for Run-time to acquaint yourself with APIs and their functioning. You can choose a Run-time sample depending on the deployment type of your application.

Deployment Types

Deployment type specifies the environment where a protected application is installed. Your application can have one of the following deployment types:

- **Cloud:** The protected application is hosted on an ISV-controlled server residing in Cloud.
- **On-premise:** The protected application is installed in a hostile environment within the customer's premises. For On-premise Deployment Type, you also need to select a Feature Caching Mode that specifies how licenses are served and cached on an on-premise machine. Following options for Feature Caching Mode are available:
 - **Entitlement Level:** This type of on-premise licensing is primarily meant for server applications. The protected application is hosted on an on-premise server machine, which is accessed by multiple thin clients. Here, an application pulls an entire entitlement and all the products/features in it.
 - **Feature Level:** This type of on-premise licensing is primarily meant for client applications. Here, an application explicitly requests the required features from Cloud Connect.

The on-premise deployment mode allows end users to use the protected application offline for a given duration.

Available Run-time Samples

For on-premise deployments, we recommend to refer to the following Run-time sample:

- [C Sample Application](#) - This C application demonstrates how to use Cloud Run-time APIs provided for C language on different operating systems.

For cloud deployments, we recommend to refer to the following Run-time samples:

- [Java Demo Application](#) - A demo application developed in Java that provides a quick walk-through of licensing, provisioning, and usage stages of delivering feature-based applications as services on the cloud.
- [.NET Reference Application](#) - This .NET application demonstrates the usage of Cloud Run-time APIs across various licensing stages, such as license availability, requesting a license, and releasing a license.

C Reference Application

Introduction

This topic provides information about the Run-time API sample available for C interface. It is useful for application developers who need to integrate the Run-time APIs into the C application source code.



For details about the Run-time APIs, refer to the [Sentinel Cloud Run-time Guide](#).

Files Included in the Sample

Windows

The Run-time C sample for Windows is available at: `<InstallDir>\Samples\C SDK`. Here, `<InstallDir>` refers to the installation directory of Sentinel Cloud Services, which by default is `C:\Program Files\Safenet Sentinel\Sentinel Cloud Services`.

The C sample includes the following:

Folder/File	Description
Demo	This folder contains the following: <ul style="list-style-type: none"> ▪ <code>src\Demo.c</code> - Main file containing API code ▪ <code>Demo.vcproj</code> - A command line based demo application that demonstrates the Run-time API usage.
<code>SentinelCloudRuntime.sln</code>	Project Solution file
<code>SentinelCloudRuntime.suo</code>	The Solution User Options (. <code>suo</code>) file
ReadMe	Contains the <i>Sentinel Cloud Run-time C Reference ReadMe</i> (this document).

Linux

The Run-time C APIs sample for Linux is available at: `<Extract Location>/SafeNet Sentinel/Sentinel Cloud Services/Samples/C SDK`. Here, `<Extract Location>` refers to the extraction directory of Sentinel Cloud Services.

Following are the contents of the `Samples` directory:

Folder/File	Description
Demo	This is the sample application folder. It contains the following: <ul style="list-style-type: none"> ▪ <code>src\Demo.c</code> - Main file containing API code. ▪ <code>Demo_x32.mak</code> - Project make file for 32-bit machine. ▪ <code>Demo_x64.mak</code> - Project make file for 64-bit machine.
ReadMe	Contains the Sentinel Cloud Run-time C Reference ReadMe (this document).

Prerequisites for Using Sample

This section contains information on what is required for using Run-time C sample.

Supported Operating Systems and Compilers

The following table provides a list of supported operating systems and compilers:

Operating System	Compiler
Windows (32-bit and 64-bit) <ul style="list-style-type: none"> ▪ XP ▪ Server 2008 ▪ Vista ▪ 7 	<ul style="list-style-type: none"> ▪ Microsoft Visual Studio 2005 ▪ Microsoft Visual Studio 2008
Linux (32-bit and 64-bit) <ul style="list-style-type: none"> ▪ Cloud <ul style="list-style-type: none"> ○ Ubuntu 10.04 ○ CentOS 5.4 ○ Red Hat Enterprise Linux (RHEL) 6.3 ▪ On-premise <ul style="list-style-type: none"> ○ Red Hat Enterprise Linux (RHEL) 6.3 	<ul style="list-style-type: none"> ▪ GCC 4.1.2

Required Libraries

The libraries required to use Run-time C APIs on different operating systems, are available at the paths specified in the table below:

Operating System	Library Path
Windows	<ul style="list-style-type: none"> ▪ For 32-bit: <InstallDir>RunTimes\C SDK\Win32 ▪ For 64-bit: <InstallDir>\RunTimes\C SDK\Win64
Linux	<ul style="list-style-type: none"> ▪ For 32-bit: <Extract Location>/ SafeNet Sentinel/Sentinel Cloud Services/Runtimes/C SDK/32-bit ▪ For 64-bit: <Extract Location>/ SafeNet Sentinel/Sentinel Cloud Services/Runtimes/C SDK/64-bit

Required Client Configuration File

The Run-time C sample uses the settings stored in the *SentinelCloudRuntime.properties* configuration file for execution.

You need to update the **DeploymentType** property in the configuration file to work on cloud/on-premise applications. This property can have **OnPremise** or **Cloud** values.

The following table lists the location of the configuration files:

Operating System	Configuration File Path
Windows	<InstallDir>\Configurations
Linux	<ExtractLocation>/SafeNet Sentinel/Sentinel Cloud Services/Configurations/

Linux Specific Requirements

- make 3.8
- For Ubuntu, the *libuuid* library that can be downloaded by using the following command:

```
sudo apt-get install uuid-dev
```


- For CentOS, e2fsprogs-devel that can be installed by using the following command

```
yum install e2fsprogs-devel
```

Required EMS Entities

Before you use the sample code, an entitlement must be created in Sentinel Cloud EMS. The typical workflow of creating an entitlement is as follows. For exact steps, refer to the *Sentinel Cloud EMS User's Guide*.



 An entitlement can be for either cloud or on-premise.

Running the C Sample

Windows

The steps to run the C sample code on Windows are:

1. Copy [the client configuration file](#) to the directory where the sample code executable resides, which is `<InstallDir>\Samples\C SDK\Demo`.

Note

If you want to place configuration file at a custom location, set the environment variable `SFNT_SCR_CONFIG_FILEPATH`.

If the `SFNT_SCR_CONFIG_FILEPATH` environment variable is not set or set to an incorrect path, the Run-time searches the configuration file at the default location.

2. Open *SentinelCloudRun-time.sln* in Microsoft Visual Studio.
3. In the *Demo.c* file, update customer, user, and vendorInfo details.
4. Open the configuration file (copied in Step 1), and update the following elements:

DeploymentType

- This parameter specifies the deployment mode to be used for compiling the application. This value can be either **OnPremise** or **Cloud**.
- Example:

```
<add key="DeploymentType" value="OnPremise" />
```

ClientAlias

- This was sent to you in a previous e-mail from SafeNet.
- This is usually the vendor name specified in the vendor registration request.
- Example:

```
<add key="ClientAlias" value="isv" />
```

YPSAddress

- This was sent to you in a previous e-mail from SafeNet.
- This parameter is the address of the Sentinel Directory Services server.
- Example:

```
<add key="YPSAddress" value="https://yps-trial.sentinelcloud.com/YPServer" />
```

CABundle

- The path of the CA bundle, which refers to the certificate store that C Run-time uses for server certificate verification.
- You can download CA certificate bundle from <http://curl.haxx.se/ca/cacert.pem>.
- Example:

```
<add key="CAbundle" value=".\cacert.pem" />
```



- If you are using proxy server, then you will need to configure the Run-time proxy settings as well. For details of proxy settings, please refer to the [Sentinel Cloud Run-time Guide](#).

- For C interface, Secret Key, Secret Key ID, Vendor ID, and Vendor Code are inserted in Run-time library during the Run-time installation.

6. Build *SentinelCloudRuntime.sln* and start the Sample Application.

Linux

The steps to run the C sample code on Linux are:

1. Copy *SentinelCloudRuntime.properties* to the directory where the sample code executable resides, which is *<ExtractLocation>/SafeNet Sentinel/Samples/Demo*.
2. For on-premise applications, specify the path in the SFNT_SCR_CONFIG_FILEPATH environment variable. The use of environment variable allows you to define the configuration settings once for the protected applications stored at different locations.

The following example shows how to set the SFNT_SCR_CONFIG_FILEPATH environment variable in */etc/profile* file on Linux:

```
SFNT_SCR_CONFIG_FILEPATH="/home/work/SaaS"
```



- If the SFNT_SCR_CONFIG_FILEPATH environment variable is not set, the Run-time looks for the configuration file in the directory where the executable exists.
- If the above environment variable is set and the configuration file is not present at the specified location, the error RT_ERR_IN_CONF_FILE_LOAD occurs.
- The user account used to execute the licensed application must have access permissions for the file path mentioned in the SFNT_SCR_CONFIG_FILEPATH environment variable.

3. Open the *SentinelCloudRuntime.properties* file, and update the following elements:

ClientAlias

- This was sent to you in a previous e-mail from SafeNet.
- This is usually the vendor name specified in the vendor registration request.
- Example:


```
<add key="ClientAlias" value="isv" />
```

YPSAddress

- This was sent to you in a previous e-mail from SafeNet.
- This parameter is the address of the Sentinel Directory Services server.
- Example:

```
<add key="YPSAddress" value="https://yps-trial.sentinelcloud.com/YPSTServer" />
```

CABundle

- The path of the CA bundle, which refers to the certificate store that C Run-time uses for server certificate verification.
- You can download CA certificate bundle from <http://curl.haxx.se/ca/cacert.pem>.
- Example:

```
<add key="CABundle" value=".\cacert.pem" />
```



If you are using proxy server, then you will need to configure the Run-time proxy settings as well. For details of proxy settings, please refer to the [Sentinel Cloud Run-time Guide](#).

4. In the *Demo.c* file, update customer, user, and feature ID details.
5. Build the Demo file by using the following command:
 - For 32-bit: `make -f Demo_x32.mak.`
 - For 64-bit: `make -f Demo_x64.mak`



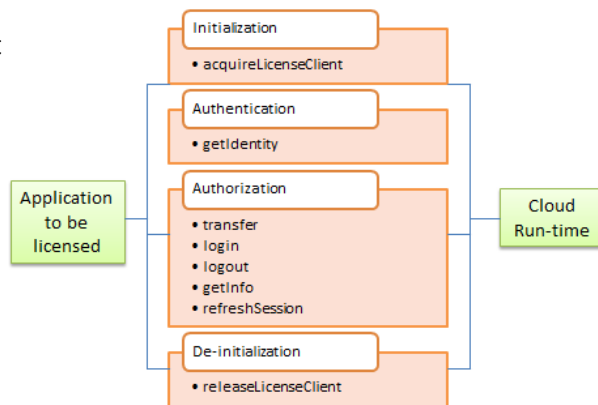
If you are using GCC 4.6 or above as the default compiler, you need to add the "**-Wl,--no-as-needed**" flag to avoid compilation errors.

6. Execute the Demo file.

Using the Run-time C APIs

Sentinel Cloud Run-time provides a set of APIs that you can use to license your applications. For API details, refer to *Sentinel Cloud Run-time Guide*.

A brief description of the APIs is given below:



Licensing Function	API	Description
Initialization	acquireLicenseClient	Obtains an instance of the Cloud Run-time.
Authentic-ation	getIdentity	Authenticates a user by requesting authentication cookie, identity, and customer reference ID from Cloud Connect. Note: This API is applicable only for on-premise feature level licensing, implemented with authentication. To use this API, an authentication system should be integrated with Sentinel Cloud. Please contact SafeNet for integration-related details.
Authorization	transfer	Detaches licenses, returns licenses, and syncs usage with Cloud Connect. Note: This API is used only for on-premise licensing.
	login	Authorizes a user by requesting a license from the Cloud Connect.
	logout	Releases the license acquired by the user.
	getInfo	Retrieves information about entitlements and features for a given user.
	refreshSession	Refreshes a concurrent session. Note: This API is used only for cloud applications.
De-ini-tialization	releaseLicenseClient	Releases the instance of the Cloud Run-time.

The *Demo.c* file contains the implementation of the Run-time APIs written in C language. Refer to this file to learn about using the Run-time APIs in your C application source code.

Java Demo Application

Introduction

The Sentinel® Cloud Demo Application is a Web application written in Java. It provides a quick walk-through of licensing, provisioning, and usage stages of delivering feature-based applications as services on the cloud.

- **As Application Developers:**
You can learn about implementing the Sentinel Cloud API calls into the application source code.
- **As Customers:**
You can learn about using the software providers' offerings as per the product models.
- **As Product Managers and Contract Operators:**
You can learn about executing the provisioning workflow in Sentinel EMS.

The following table lists the products offered by the Demo Application:

Products	Features	Corresponding Feature IDs
Calculator	<ul style="list-style-type: none">■ Addition■ Subtraction■ Multiplication■ Division	<ul style="list-style-type: none">■ 1■ 2■ 3■ 4
Stopwatch	Stopwatch	5
Media Player	Media Player	6



The Demo Application can be deployed on cloud as well as on customer's premises.

Installing and Configuring Java Demo Application

System Requirements

The software requirements to install Demo Application are:

- Java Run-time Environment (JRE 6)
- JDK 1.6
- A Web Server, such as Apache Tomcat 7.0, that can run a Java Web application.



The on-premise Linux application running on Tomcat should be able to read the LC_CTYPE environment variable, which stores system locale.

Installation and Configuration



The steps given below are for Tomcat 7.0.

1. Copy the file *SaaS DemoApplication.war* from the installation media to the `<tomcat_home>/tomcat7/webapps` directory on your machine.
Here, `<tomcat_home>` is the Tomcat installation directory.
2. Start your Web Server.
3. Go to `<tomcat_home>/tomcat7/webapps/SaaS DemoApplication/WEB-INF/classes`, and open the *SentinelCloudRuntime.properties* configuration file. You need to use this configuration file for deploying the Cloud or on-premise applications.



For on-premise applications, the location where the configuration file is placed must be set in the SFNT_SCR_CONFIG_FILEPATH environment variable. The on-premise application will not work if the configuration file is placed in a jar at the class path.

4. In the client configuration file, update the following elements:

DeploymentType

- This parameter specifies the deployment mode to be used for compiling the application. This value can be either **OnPremise** or **Cloud**.
- Example:

```
<add key="DeploymentType" value ="Cloud" />
```

SecretKey

- This was sent to you in a previous e-mail from SafeNet.
- This is the secret key used for authenticating client requests.

- Example:

```
<add key="SecretKey" value="SecretKey" />
```

SecretKeyId

- This was sent to you in a previous e-mail from SafeNet.
- This is the ID of the secret key.
- Example:

```
<add key="SecretKeyId" value="SecretKeyId" />
```

VendorID

- This was sent to you in a previous e-mail from SafeNet.
- This parameter is the unique ID assigned to a vendor by SafeNet.
- Example:

```
<add key="VendorId" value="a8e06c3" />
```

ClientAlias

- This was sent to you in a previous e-mail from SafeNet.
- This is usually the vendor name used in the vendor registration request.
- Example:

```
<add key="ClientAlias" value="isv" />
```

YPSAddress

- This was sent to you in a previous e-mail from SafeNet.
- This parameter is the address of the Sentinel Directory Services server. Set `YPSAddress` correctly to ensure that the Demo Application works.
- Example:

```
<add key="YPSAddress" value="https://yps-trial.sentinelcloud.com/YPServer" />
```



If you are using proxy server, then you will need to configure the Cloud Run-time proxy settings as well. For details of proxy settings, please refer to the [Sentinel Cloud Run-time Guide](#).

5. For on-premise applications, copy `SentinelCloudRuntime_P.dll` (32-bit) or `SentinelCloudRuntime_P_x64.dll` (64-bit), at:
 - `C:\WINDOWS\system32`
 - a custom location and set the location in the **Path** environment variable.

6. Restart the Web Server.

Accessing the Java Demo Application

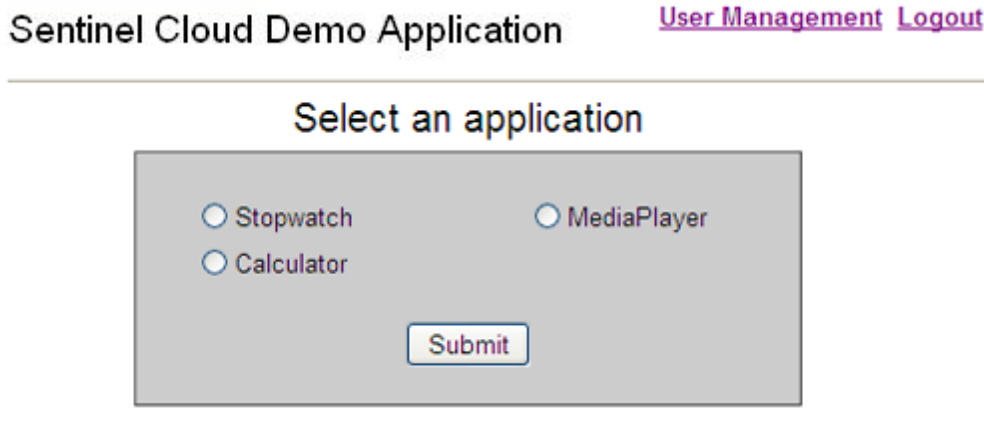
After installation and configuration, you can access the Cloud Demo Application by using the following URL:

http://localhost:8080/SaaS Demo Application

Here, *localhost* is the IP address or domain name of the host. Following are the login credentials for Demo Application:

- **User Name:** admin
- **Password:** admin

Java Demo Application User Interface



Sentinel Cloud Demo Application [User Management](#) [Logout](#)

Select an application

Stopwatch MediaPlayer

Calculator

In the above interface, the **User Management** link is enabled for only the “admin” user.

See also: [Creating Users for the Cloud Demo Application](#).

Before You Use the Java Demo Application...

Before a user consumes the Demo Application products, the following steps need to be performed:

- [Creating Entitlements in the Sentinel EMS](#)
- [Creating Users for the Cloud Demo Application](#)

Creating Entitlements in the Sentinel EMS

Before using the Java Demo Application, you need to have an understanding of how to create feature, product, and entitlement in EMS.

The typical workflow of creating an entitlement is as follows. For exact steps, refer to the *Sentinel Cloud EMS User's Guide*:



- An entitlement can be for either cloud or on-premise.
- Entitlements created for the Java Demo Application must use same feature information, especially feature ID, as mentioned in the section [Introduction](#).
- While the entitlements generated can be used right away by users, data aggregation begins only after an hour of the deployment. Beyond this, data aggregation will take place as per the frequency set in EMS.

Creating Users for the Cloud Demo Application

Initially, a user with name "admin" and password "admin" can login into the hosted Demo Application. Only this user has rights to add further users. The steps of adding users are described below:

1. Click the **User Management** link on the top-right corner of the page. The User Details screen is displayed.
2. From the **Customer** drop-down list, select the customer if it already exists. Otherwise perform the following steps to create a new customer:
 - a. Click the **Create Customer** link. The Customer Details page is displayed.

- b. Enter the customer name and customer reference ID. These should be same as specified in EMS.
 - c. Click the **Add** button. The customer is added and the User Details screen is displayed again.
3. To add a new user for the selected customer, provide the user name and password, and click the **Add** button.



The Customer and User created in Demo Application should be exactly same as specified in EMS entitlement. For example, if a customer name is "Frank Duff" in EMS but you have used "Frank" in Demo Application, an error will be thrown.

Using the Java Demo Application Products

Once the users are added into the system, they can access the licensed products using their login name and password. An error will be thrown if any feature is not allowed.

See also, [Error Messages](#).

Viewing Usage Related Information

Users can login into Sentinel EMS (Customer Login) using their entitlement IDs (received in the e-mail), and can view the following information:

- Usage details of an entitlement (available only if the entitlement has been aggregated by Data Engine).
- Billing detail
- License state of the features
- Concurrent sessions for features having concurrent license model
- Usage count of features having pre-paid license model

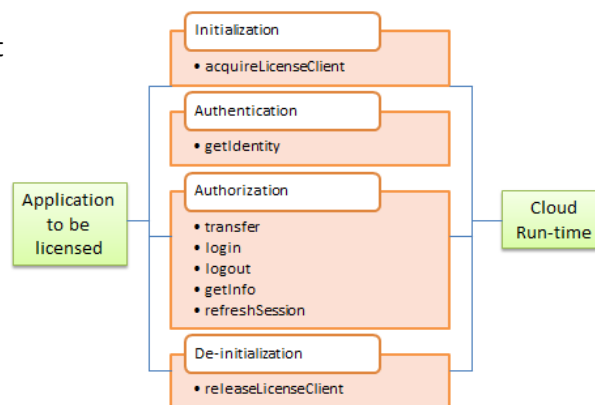


For more information, please refer to *Sentinel Cloud EMS User's Guide*.

Using the Sentinel Cloud Run-time Java APIs

Sentinel Cloud Run-time provides a set of APIs that you can use to license your applications. For API details, refer to *Sentinel Cloud Run-time Guide*.

A brief description of the APIs is given below:



Licensing Function	API	Description
Initialization	acquireLicenseClient	Obtains an instance of the Cloud Run-time.
Authentic-ation	getIdentity	Authenticates a user by requesting authentication cookie, identity, and customer reference ID from Cloud Connect. Note: This API is applicable only for on-premise feature level licensing, implemented with authentication. To use this API, an authentication system should be integrated with Sentinel Cloud. Please contact SafeNet for integration-related details.
Authorization	transfer	Detaches licenses, returns licenses, and syncs usage with Cloud Connect. Note: This API is used only for on-premise licensing.
	login	Authorizes a user by requesting a license from the Cloud Connect.
	logout	Releases the license acquired by the user.
	getInfo	Retrieves information about entitlements and features for a given user.
	refreshSession	Refreshes a concurrent session. Note: This API is used only for cloud applications.
De-ini-tialization	releaseLicenseClient	Releases the instance of the Cloud Run-time.

To learn about how to use Cloud Run-time Java APIs in your source code, refer to the files *License.java* and *ContextLoaderListener.java*, available at: `\src\com\sfnt\saas\demo\`.

- *License.java*: Contains implementation of the login and logout APIs.
- *ContextLoaderListener.java* : Contains implementation of the acquireLicenseClient and releaseLicenseClient APIs.

Error Messages



The error messages described in this section are specific to the Cloud Run-time Java Demo Application. For information on other Run-time errors, please refer to [Sentinel Cloud Run-time Guide](#).

Error Message	Description
User does not exist	User name is invalid.
Either user name or password is incorrect	User name or/and password are invalid.
License does not exist or license is not in active state	Reason can be: <ul style="list-style-type: none"> ▪ No entitlement/product catalog exists for requested feature and user. ▪ The feature start date has not reached yet.
License is expired	The feature end date has reached already.
License is disabled	The software publisher has disabled the contract/product catalog.
Failed to create database connection	The database server is down. Contact the SafeNet Technical Support for assistance.
Access denied to the requested feature	The user is not authorized for the requested feature.
Connection to Yellow Pages Server failed	The connection to Directory Services server is not established. Try the following to resolve this error: <ul style="list-style-type: none"> ▪ Ensure that <i>YPSAddress</i> is specified correctly in the client configuration file. ▪ Check logs at the path given in <i>log4j.properties</i> file in the <i>\WEB-INF</i> directory for more details.

Building the Cloud Java Demo Application

Given below is the directory structure of the Demo Application source code available at:
<InstallDir>\Samples\Java SDK:

Here, <InstallDir> refers to the installation directory of Sentinel Cloud Services, which by default is
C:\Program Files\Safenet Sentinel\Sentinel Cloud Services.

Directory	Contents
<i>build</i>	Contains the ant file used for building the Demo Application.
<i>settings</i>	Contains Eclipse project settings.
<i>lib</i>	Contains all dependent libraries.
<i>resources</i>	Contains all resource files, including the Cloud Run-time resource files.
<i>src</i>	Contains source code.
<i>WebContent</i>	Contains the following Web contents: <ul style="list-style-type: none"> ▪ <i>css</i>: Contains style sheet files. ▪ <i>Images</i>: Contains required images. ▪ <i>META-INF</i>: Contains manifest file. ▪ <i>WEB-INF</i>: Contains <i>web.xml</i> and <i>log4j</i> files. ▪ <i>Various JSP</i> and miscellaneous files.
<i>ReadMe</i>	Contains the reference document for Java Demo Application (this document).

You can build the Demo Application by using any of the following methods:

- [Using Eclipse](#)
- [Using the Ant Build Script](#)

Using Eclipse

Prerequisites

- Run the ant script.

Follow the steps given below to build the Demo Application in Eclipse:

1. Click **File > Import**.
2. Select **Existing Projects into Workspace** under **General**. Click **Next**.
3. Browse to the SaaS Demo Application directory in source and click **OK**.
4. Click **Finish**.

Add *jar* files from the *\lib* folders, by using the following steps:

- a. Click **Project >Properties**.
- b. In the Properties dialog box, click **Java Build Path** in the left pane, and then click the **Libraries** tab.
- c. Click the **Add External Jars** button, browse and select the jar files, and click **OK**.

Using the Ant Build Script

Follow the steps given below to build the Demo Application using the Ant script:

1. Go to the command prompt.
2. Go to the `<InstallDir>\Samples\Java SDK\build` directory.
3. Type `ant` and press **Enter**. It will create a `SaaS DemoApplication.war` file in the `<InstallDir>\Samples\Java SDK\target\War` directory.

Notes:



- Make sure Apache Ant is installed on the system and ANT_HOME environment variable is set.
- Make sure JDK 1.6 or higher version is installed and JAVA_HOME environment variable is set.
- This build procedure is tested with Apache Ant 1.8.1 version.

Frequently Asked Questions - Java Demo Application

Question: How to uninstall the Cloud Demo Application?

Delete the installed directory `\SaaS Demo Application` and its `WAR` file from `/var/lib/tomcat7/webapps`.

Question: How to change the admin default password?

To change the admin's default password, edit the `SaaS Demo.xml` file present in `var/lib/tomcat7/webapps/SaaS Demo Application/WEB-INF/classes` with new password as shown below:

```
<Customer Name=""admin"">
<User>
<User ID>admin</User ID>
<UserPassword>admin</UserPassword>
</User>
</Customer>
```

.NET Reference Application

Introduction

The Sentinel® Cloud Run-time .NET Reference Application demonstrates the usage of Cloud Run-time APIs. It is useful for application developers who need to implement the Cloud Run-time API calls into their .NET application source code.

The .NET Reference Application covers the following stages of licensing:

Licensing	Usage
License Availability	Checks if a license can be granted to user or not.
Requesting a License	Authorizes a user by requesting a license from the Cloud Connect.
Releasing a License	Releases the license acquired by the user.



For details about the Cloud Run-time APIs, refer to [Sentinel Cloud Run-time Guide](#).

Location of .NET Reference Application

The Reference Application is available after you install Sentinel Cloud Services .NET SDK. It is present in the installation directory at the path: `<InstallDir>\Samples\Dot NET SDK`.

Here, `<Install-Dir>` refers to the installation directory of Sentinel Cloud Services, which is `C:\Program Files\SafeNet Sentinel\Sentinel Cloud Services\`, by default.



For information about .NET SDK installation, refer to *Sentinel Cloud Services Installation Guide*.

Software Requirements

The following table lists the software requirements for configuring and using the .NET Reference Application.

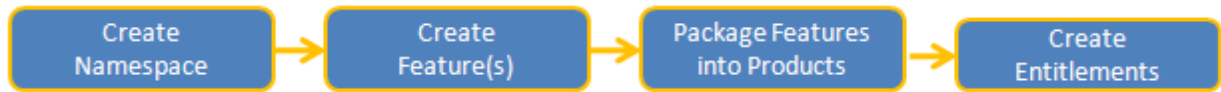
Software	Tested Versions
Windows Operating Systems	Windows Server 2008
Microsoft .NET Framework	2.0, 3.5
Microsoft Visual Studio	2005
Web Server	IIS 7.0
Web Browser (Recommended)	Internet Explorer 7



The .NET Application can be run with other software versions as well, however, only the tested versions are listed above.

Before You Use the .NET Reference Application...

Before you use the .NET Reference Application, entitlements must have been created in the Sentinel EMS. The typical workflow of creating an entitlement is depicted below. For exact steps, refer to the *Sentinel Cloud EMS User's Guide*:



- An entitlement can be for either cloud or on-premise.
- The “License availability” and “Request a License” features of the .NET Reference Application must use the entitlement information created in EMS.
- While the entitlements generated can be used right away by users, data aggregation begins only after an hour of the deployment. Beyond this, data aggregation will take place according to the frequency defined in EMS.



Building the .NET Reference Application

1. Navigate to the folder where the [.NET Reference Application is installed](#).

The default path is: *C:\Program Files\Safenet Sentinel\Sentinel Cloud Services\Samples\Dot NET SDK*.

2. Copy all the files from the installation folder to a folder on your machine, say **\DevCode\Sentinel Cloud Samples**.

Example – Copy from: *\Samples\Dot NET SDK*

to: *\DevCode\Sentinel Cloud Samples*

3. Add references of the following library files, which are available at *C:\Program Files\Safenet Sentinel\Sentinel Cloud Services\RunTimes\Dot NET SDK\External Libraries*:

- BouncyCastle.Crypto.dll
- Common.Logging.dll
- log4net.dll
- Microsoft.Practices.EnterpriseLibrary.Caching.dll
- Microsoft.Practices.EnterpriseLibrary.Common.dll
- Microsoft.Practices.ObjectBuilder.dll
- Quartz.dll
- Quartz.xml

These files will be copied to: *\DevCode\Sentinel Cloud Samples\Bin*.

4. Add reference of the following DLL file, which is available at *C:\Program Files\Safenet Sentinel\Sentinel Cloud Services\RunTimes\Dot NET SDK\Library* :

- *SentinelCloudRuntime.dll*: For cloud applications

This file will be copied to: *\DevCode\Sentinel Cloud Samples\Bin*

5. For On-premise applications, copy *SentinelCloudRuntime_P.dll* (32- bit) or *SentinelCloudRuntime_P_x64.dll* (64- bit), at:
 - C:\WINDOWS\system32
 - A custom location and add the location in the **Path** environment variable.
6. Open Microsoft Visual Studio. Go to **File > Open WebSite** and open the sample. In our example, open *C:\DevCode\Sentinel Cloud Samples*.
7. Open the **Web.Config** file and copy its sections in your application configuration file.
8. For on-premise deployment, set the path of your application configuration file in the SFNT_SCR_CONFIG_FILEPATH environment variable.
9. Update the following elements in the configuration file:

DeploymentType

- This parameter specifies the deployment mode to be used for compiling the application. This value can be either **OnPremise** or **Cloud**.

- Example:

```
<add key="DeploymentType" value="Cloud" />
```

ClientAlias

- This was sent to you in a previous e-mail from SafeNet.
- This is usually the vendor name specified in the vendor registration request.
- Example:

```
<add key="ClientAlias" value="isv"/>
```

YPSAddress

- This was sent to you in a previous e-mail from SafeNet.
- This parameter is the address of the Sentinel Directory Services server.
- Example:

```
<add key="YPSAddress" value="https://yps-trial.sentinelcloud.com/YPSTServer" />
```

SecretKey

- This was sent to you in a previous e-mail from SafeNet.
- This is the secret key used for authenticating client requests.
- Example:

```
<add key="SecretKey" value="SecretKey" />
```

SecretKeyId

- This was sent to you in a previous e-mail from SafeNet.
- This is the ID of the secret key.
- Example:

```
<add key="SecretKeyId" value="SecretKeyId" />
```

VendorID

- This was sent to you in a previous e-mail from SafeNet.
- This parameter is the unique ID assigned to a vendor by SafeNet.
- Example:

```
<add key="VendorId" value="a8e06c3" />
```



If you are using proxy server, then you will need to configure the Cloud Run-time proxy settings as well. For details of proxy settings, please refer to the [Sentinel Cloud Run-time Guide](#).

10. In Visual Studio, set the start page to **getInfo.aspx**.
11. Build and start the Web site.

Accessing the .NET Reference Application

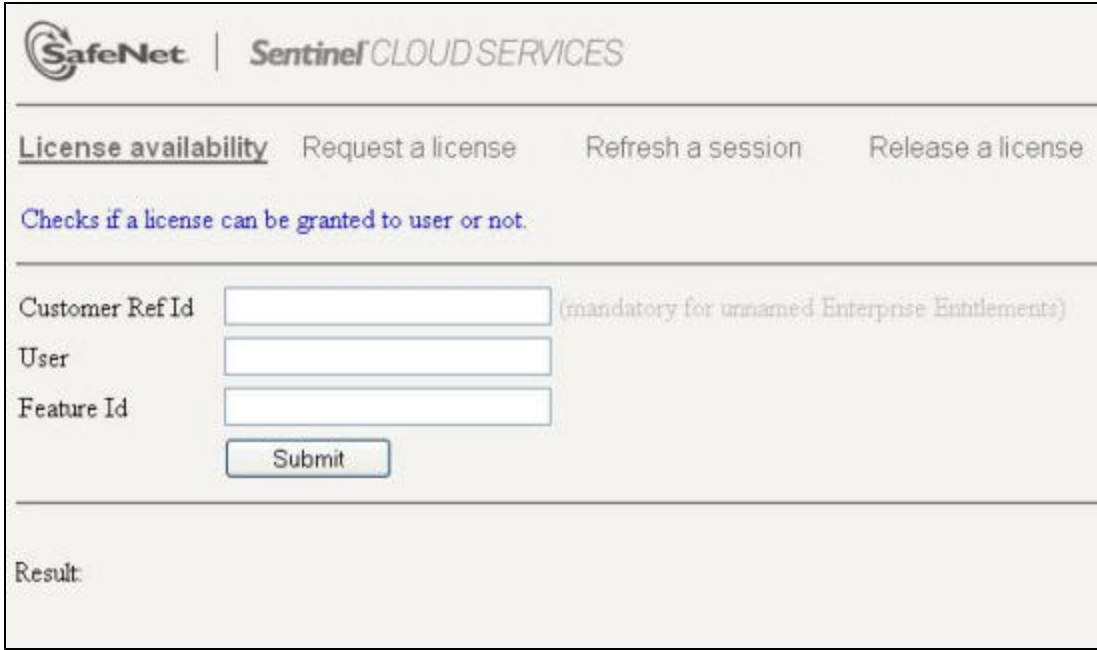
After building the .NET Reference Application, you can access it by using the following URL:

http://localhost:portnumber/Sentinel Cloud Samples/login.aspx.

In this URL:

- Localhost is the IP address of the host
- Port number is the port used by the application

.NET Reference Application User Interface



The screenshot shows the user interface for the Sentinel Cloud Services application. At the top, there is a header with the SafeNet logo and the text "Sentinel CLOUD SERVICES". Below the header, there are four links: "License availability", "Request a license", "Refresh a session", and "Release a license". Under "License availability", there is a sub-link that says "Checks if a license can be granted to user or not." Below this, there are three input fields: "Customer Ref Id" (with a note "(mandatory for unnamed Enterprise Entitlements)"), "User", and "Feature Id". A "Submit" button is located below the "Feature Id" field. At the bottom of the form, there is a "Result:" label.

 We assume that the Sentinel EMS Web portal is already up and running.

The Cloud Run-time Reference Application can be hosted on the IIS server using the **Build > Publish Web site** option in Microsoft Visual studio.

Using the .NET Reference Application

Once entitlements are created into the system, a user can access the .NET Reference Application and see the usage of APIs.



An error will be thrown if any feature is not allowed. For a list of possible error codes, refer to the [Sentinel Cloud Run-time Guide](#).

Viewing Usage Related Information

Users can login into Sentinel EMS (Customer Login) using their entitlement IDs (received in the e-mail), and can view the following information:

- Usage details of an entitlement (available only if the entitlement has been aggregated by Data Engine).
- Billing detail
- License state of the features
- Concurrent sessions for features having concurrent license model
- Usage count of features having pre-paid license model

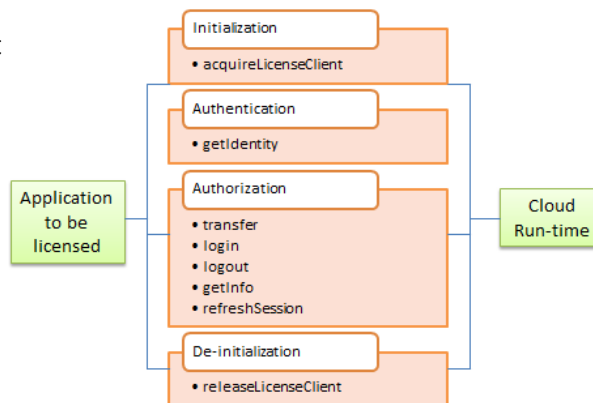


For more information, please refer to *Sentinel Cloud EMS User's Guide*.

Using the Sentinel Cloud Run-time .NET APIs

Sentinel Cloud Run-time provides a set of APIs that you can use to license your applications. For API details, refer to *Sentinel Cloud Run-time Guide*.

A brief description of the APIs is given below:



Licensing Function	API	Description
Initialization	acquireLicenseClient	Obtains an instance of the Cloud Run-time.
Authentic-ation	getIdentity	Authenticates a user by requesting authentication cookie, identity, and customer reference ID from Cloud Connect. Note: This API is applicable only for on-premise feature level licensing, implemented with authentication. To use this API, an authentication system should be integrated with Sentinel Cloud. Please contact SafeNet for integration-related details.
Authorization	transfer	Detaches licenses, returns licenses, and syncs usage with Cloud Connect. Note: This API is used only for on-premise licensing.
	login	Authorizes a user by requesting a license from the Cloud Connect.
	logout	Releases the license acquired by the user.
	getInfo	Retrieves information about entitlements and features for a given user.
	refreshSession	Refreshes a concurrent session. Note: This API is used only for cloud applications.
De-ini-tialization	releaseLicenseClient	Releases the instance of the Cloud Run-time.

To learn about using the Sentinel Cloud Run-time .NET APIs in your source code, refer to the files *login.aspx.cs*, *logout.aspx.cs* and *getInfo.aspx.cs*, available at: `\Samples\Dot NET SDK`.

File	Purpose	Returns
<i>login.aspx.cs</i>	Contains a sequence of APIs to implement the login API. This file is corresponding to the “request a license”	Successful execution of the login API returns a session handle, which is displayed on the Release

File	Purpose	Returns
	stage.	a License page. The inputs displayed on the screen are parameters of the login API.
<i>logout.aspx.cs</i>	Contains a sequence of APIs to implement the logout API. This file is corresponding to the "release a license" stage.	Session handle obtained in the above step should be used here to release license. The inputs displayed on the screen are parameters required by the logout API.
<i>getInfo.aspx.cs</i>	Contains a sequence of APIs to implement the getInfo API, which helps check the feature authorization. This file is corresponding to the "license availability" stage.	A list of entitlements is returned by getInfo. Search this list to check the feature availability, and set the value of "Result" based on the license status.
<i>refreshSession.aspx.cs</i>	Contains a sequence of APIs to implement the refreshSession API. This file is corresponding to the "refresh a session" stage.	Session handle to be refreshed. The session handle is same as the one returned by the login API.



"Exception" and "Error code" display any exception and error code, respectively, as returned by the Cloud Run-time APIs.

Commonly Used Terms

Attribute

Properties that make a license, for example, Start Date, End Date, and Concurrent Limit. By using different combination of attributes, a variety of license models can be generated.

Authorization

Controls user access to a feature as defined by the service agreement and the governing license model.

Authorization Map

The set of authorized features per user.

Capacity

Indicator of load on a feature during a login session. The load can have different meanings in different environments. It could be number of users consuming the application at a given time or the number of cores on which the application is running.

Capacity Attribute

A license attribute that is used to enable processing of capacity values recorded by an application for postpaid features. You must set Capacity Attribute in EMS if you want to be notified when your customers approach or reach a specified limit.

Catalog

The collection of Namespaces, Products, Features, License Models, and Service Agreements available in EMS.

Cloud Connect

Component that authorizes the feature requests coming from Cloud Run-time instances. It is also responsible for activating the customer service agreement and enforcing license models.

Contact

A single point of contact for an entitlement who is identifiable with a unique e-mail address. All the entitlement related e-mails are sent to the e-mail address specified for the contact.

Concurrency

A license property that allows wider access to a feature but limits the number of simultaneous instances that can be run of the feature.

Concurrent Limit

Maximum number of concurrent instances allowed for a feature. The concurrent instances can be counted Per Login or Per Identity.

Customer

An individual or enterprise who places an order for your Products.

Customer Administrator

Role assigned to an EMS user that grants the user the right to perform certain administrative functions. Customer Administrator views customer entitlements, usage information, and current state of licenses. This role also manages Enterprise User IDs.

A Customer Administrator and Contact basically do the same functions.

Data Aggregation

The process of converting raw usage data into processed data that can be used for billing and metering.

The raw usage records are collected by Sentinel Cloud Run-time and are pushed to Sentinel Cloud Connect periodically. Sentinel Cloud Connect aggregates the multiple records into one single record in every 24 hours. The aggregated records form the basis for generating bills, and are also used in EMS to present usage and billing data to ISV/customer.

Data Aggregation can be Count Based or Time Based.

Data Aggregation Frequency

The time interval after which the usage data is aggregated.

Data Engine

Component that stores, processes, and aggregates usage records.

Developer

Role assigned to an EMS user that grants the user the right to perform activities related to preparation of a protected application, such as planning and implementing the licensing policy for their software applications.

Also called: ISV, Vendor, Software Publisher, Service Provider, Software Provider.

Deployment Type

Specifies if the entitlement is created to license the applications deployed on-premise or on cloud.

Directory Services

Component that manages the list of registered Sentinel Cloud Connect and Sentinel Cloud Run-time instances. Directory Services determines which Cloud Connect nodes serve a Cloud Run-time.

Disconnect Limit

Number of days for which the on-premise application can remain disconnected from Sentinel Cloud Connect.

End Date

License Attribute that restricts use of a feature till a specified date.

End Date can be set to 'No Limit' or a to very large value to create Perpetual licenses models.

EMS (Entitlement Management System)

A Web-based solution that provides a centralized interface for all license and entitlement management functions.

EMS Administrator

EMS user who performs administrative functions, such as defining and managing namespaces, EMS users, and roles.

EMS User

A person who has credentials to log in to EMS.

End User

The actual consumer of the feature. It can be an individual customer or member of an enterprise on whose computer the licensed application is installed of the product an ISV is selling in the market.

Entitlement

The right of a customer to use a protected application in accordance with the terms of the license.

Entitlement Type

Type of entitlement, which can be:

- **Retail:** Entitlements created for an individual. A retail entitlement is always 'Named'.
- **Enterprise:** Entitlements created for an organization having multiple users. It can be either 'Named' or 'Unnamed'.

Enterprise Entitlement

See description of "Entitlement Type" above.

Entitlement Manager

EMS role that defines, manages, and provisions customer orders.

Executions

Term used in context of Prepaid license models. It allows a specified number of executions of a feature.

Feature

An identifiable functionality of a software application that can be independently controlled by a license. A Feature may be an entire application, a module or a specific functionality such as Print, Save, or Draw.

Feature ID

A unique identifier for a feature. It helps enforce authorization at the feature level.

Globally Concurrent License

Licenses (like pre-paid and concurrent licenses) which need synchronization across the Cloud Run-time nodes. License information is not cached for globally concurrent licenses.

Grace Limit

The additional number of days/times the feature can be used after its license has expired/exhausted.

Identity

Represents a Named User.

Instance Counting

Attribute that defines the criteria of counting concurrent instances.

License

A collection of attributes relating to a Feature that indicate how the user is allowed to use the Feature within a protected application.

License Model

A combination of license attributes that controls how a feature is used. It represents a licensing strategy.

Maximum Count

The maximum number of times a Prepaid feature can be used.

Namespace

A logical partition for the EMS workspace. Features and Products are assigned to specific namespaces. If you have different product lines, you can maintain different namespaces for them.

Named Entitlement

An Entitlement (Retail/Enterprise) that allows access to a feature by a specific number of Named users.

On-premise

A deployment type where the protected application is hosted within the premises of an enterprise customer.

Peak Capacity

The cumulative value of capacity. It indicates the cumulative load across a number of simultaneous sessions that are in use by a customer.

Product

The item that a software provider offers to sell. A product bundles features, each of which is associated with one or more license model. In addition, a product is associated with a Service Agreement.

Product Manager

Role assigned to an EMS user, usually a product manager, who determines the Features to be protected, how these Features are used to define Products, and the license structure for the Products.

Protected Application

The ISV software product that is protected and licensed with Sentinel.

Reconfiguration

The process of updating a committed entitlement.

Reports

EMS generated reports containing valuable business information from the EMS database.

Retail Entitlement

Entitlements created for an individual. A retail entitlement is always 'Named'.

See Also: Entitlement Type, Named Entitlement, and Unnamed Entitlement.

Role

Definition of a set of authorizations for a function performed by an EMS user.

Run-time

Component that integrates with the Web application to be protected with Sentinel. As the users access the application over the cloud, the first interface is with the Sentinel Cloud Run-time.

Run-time APIs

A set of licensing APIs provided by Cloud Run-time. These are inserted in application source code to enable user authorization, licensing, and usage data collection.

Service Agreement

A mutual agreement between a software provider and a customer that usually defines various attributes of products, such as level of quality, performance, availability, etc. Currently, Service Agreement defines Data Aggregation Frequency of a Product.

Every product must have a service agreement associated to it.

Unnamed Product

A type of Enterprise Entitlement, in which any number of enterprise users can use the products. There is no need to provide any specific User ID.

See Also: Named Entitlement, Unnamed Entitlement, Enterprise Entitlement

Usage Information

Usage data of a Product by each Customer, at the feature level. Sentinel Cloud Run-time stores this data locally and periodically transfers it to the Sentinel Cloud Connect for metering and data aggregation purposes. The aggregated data is later used for billing.

Related Terms: Data Aggregation, Usage type

Usage Type

Defines type of data aggregation.

- **Time Based:** Duration of each login session is aggregated. Billing is done based on the duration of period for which the feature is used.
- **Count Based:** Each login-logout pair is counted as one execution. Billing is done based on the number of executions within the specified period.

User ID

Identifies a User of a Named Entitlement (Retail or Enterprise).

Vendor ID

Identifies the vendor or ISV.

Obtaining Support

You can contact us using any of the following options:

- **Business Contacts** - To find the nearest office or distributor, use the following URL:
<http://www.safenet-inc.com/contact-us/>
- **Technical Support** - To obtain assistance in using SafeNet products, feel free to contact our Technical Support team:
 - Phone: 800-545-6608 (US toll free), +1-410-931-7520 (International)
 - E-mail: support@safenet-inc.com
 - URL: <http://sentinelcustomer.safenet-inc.com/sentinelsupport/>
- **Downloads** - You may want to check out updated installers and other components here:
www.sentinelcustomer.safenet-inc.com/sentineldownloads/

