Point SDK Technical Integration Guide

point

December 2021.

6475 Camden Avenue, Suite 207

San Jose, CA 95120

Corporate Office: (408) 997-5525

Dedicated Customer Service: (800) 342-2599

Sales: (800) 362-2599

Document Revisions

Version	Version	Document Changes
11.2	Point Requirements	Updated Point requirements

Table of Contents

List of Tables

List of Figures

Preface

The Point Software Development Kit (Point SDK™) application accesses Point data for specific needs and environment. The SDK Technical Integration guide is a high-level overview of the Point SDK. Software developers use the guide to evaluate the Point SDK to determine if the application is compatible with company business requirements.

Applications developed with the Point SDK can:

- Access Point loan data from local folders or from PointCentral®.
- Run reports on loan data using the Point report templates.

Point SDK users and vendors are not impacted if Calyx Software® changes the Point data format or encrypts data.

The Point SDK is available as a .NET assembly DLL compiled under .NET Framework 4.8. Develop your applications with .NET Framework version 4.8 to ensure compatibility.

Product Assistance and Support

Calyx Software provides free support for all products and services. Use the following contact information for sales and technical support:

Table 1: Product Assistance and Support

Point SDK Support	**Phone:** 1 (800) 342-2599 **Email:** dcsonline@calyxsoftware.com **Website:** www.calyxsoftware.com

Additional Resources

The following links provide additional resources for third party software products:

Refer to the Microsoft .NET Framework Developer's Guide for additional information about creating a key pair or the Strong Name tool, sn.exe:

- Creating a Key Pair
- Strong Name Tool sn.exe

Refer to the following Microsoft .NET development class libraries documentation for more information about folders searched in the advanced deployment scenario:

- AppDomain Class
- AppDomain.BaseDirectory Property
- AppDomain.BaseDirectory Property

- [AppDomain.RelativeSearchPath Property](#)

Refer to the following Microsoft documentation for additional information about signing .NET assemblies with a strong name and additional options such as delay signing:

- [Sign an assembly with a strong name](#)
- [Delay signing an assembly](#)

Development Requirements

Use development tools that are compatible with .NET 4.8. The Point SDK development system requires the following software.

Table 2: Development Requirements

Component	Version
Calyx Point	Version 10.3, or later
Development tool	Development tools compatible with .NET 4.7.2

Point Requirements

System Requirements

The following system requirements are required for Point installations.

Table 3: System Requirements

Component	Version
Processor speed	800 MHz (or higher)
Memory (RAM)	2 GB, minimum
Disk space	- Point application: 300MB - Microsoft .NET Framework®: up to 4.5GB - Electronic document storage: 20MB per loan file (on average

Software Requirements

The following software is required to use Point.

Table 4: Software Requirements

Component	Version
Operating System	One of the following operating systems is required: • Microsoft Windows® 11 • Microsoft Windows® 10 • Microsoft® Server 2012 or higher
Web browser	Microsoft Edge®
Microsoft .NET Framework®	Version 4.8
Modem/Internet Connectivity	An active Internet connection is required to install Point and access Internet-enabled services in Point.
Other	• Adobe Acrobat Reader© Version 11.0 DC, or later, is required for reading PDF files. • Microsoft Office® 2016 and 2019

Forward Compatibility

The Point SDK is intended to work with the version of Point and the Point SDK that code is compiled against and future versions of Point and the Point SDK. Calyx Software provides advance notice before the final SDK is released when there is a change between versions of Point and the Point SDK.

Installation

The Point SDK is automatically installed during the Point application installation. The installation process is different for different types of SDK subscribers, such as: vendor or user.

The Point SDK must be activated before you can begin development and at application runtime.

Vendor SDK Installation

The *Account ID* used for the Point installation is the same *Account ID* that is used to activate the Point SDK.

User SDK installation

Point installation for the user SDK is performed from the MyCalyx portal on the Calyx Software website. The installation email containing installation instructions is sent when users are set up on the MyCalyx website.

Go to the *MyCalyx Help* page for information about how to create a MyCalyx Administrator or a MyCalyx account.

Refer to the **Adding Users** section on the *MyCalyx Help* page for information about adding Point users in MyCalyx.

Use the following instructions to install Point after receiving the email.

1. Select the **Install Point** link from the received email.
2. Select the checkbox to accept the terms of the agreement.

Figure 1: SDK installation

3. Click **Install**.

The installation is completed per the settings in the assigned installation group. The default templates and data folders are created when an installation group is not assigned.

Activation

The Point SDK activation in the development environment must be completed before using Point SDK.

The following are required to activate the Point SDK:

- An active internet connection.
- A firewall that allows outgoing SSL connections on port 443: Outgoing connections go to https://tmsservice.calyxsoftware.com/ during runtime.
- A public and private key token: Point SDK uses the public key token from assemblies signed with a strong name to identify the software publisher. The public key token determines whether the software publisher can use and deploy the Point SDK after the key is identified.

Calyx Software supplies the batch file **createmykeys.bat** in **C:\WINPOINT\sdk** if a key pair is not already available to simplify the process.

Run the batch file from the *Microsoft Visual Studio 2017* command prompt: **Microsoft Visual Studio 2017** →**Developer Command Prompt for VS 2017** to generate the following key files in the **C:\WINPOINT\sdk** directory.

Note: Versions higher than 2017 are compatible.

Table 5: SDK Keys

File name	Description
MyPrivateKey.snky i	Use to sign assemblies with a unique strong name
MyPublicKey.snk	Distributed with assemblies that are signed with a strong name
MyPublicKeyToken.txt	Public key token used to activate and use the Point SDK.

Email the token with company name and account ID to Calyx at SDK@calyxsoftware.com after creating the public key token.

Point SDK Support performs the following after receiving the public key token:

- Logs the key token in the Calyx servers.
- Provides the mypdklicense.lic license file.
- Provides notification when activation is complete.

Advanced Deployment

The Point SDK searches for the license file in the following folder order during advanced deployment scenarios.

Table 6: Advanced Deployment

Design mode	Runtime
When working in Microsoft Visual Studio .NET and adding the Point SDK DLL to a project:	When deploying and running applications from your installation directory:
- Project folder - Project\Bin folder for Web projects. - Project build output folder. - Project debug working folder. - Location of Point SDK DLL assembly reference in Microsoft Visual Studio. .NET	- CodeBase of Point SDK DLL assembly. - CodeBase of your project assembly (EXE or DLL). - BaseDirectory of current AppDomain. - RelativeSearchPath of current AppDomain. - ApplicationBase of current AppDomain. - PrivateBinPath of current AppDomain. - PrivateBinPathProbe of current AppDomain.

The Point SDK searches folders and evaluates located license files. Licenses are evaluated alphabetically in the searched folder using the sort order defined by the current culture settings in the operating system.

Runtime Activation

The Point SDK runtime activation is performed the first time the application is run on a single installation.

The Point SDK first searches for the license in thefolder where the application is deployed during the runtime activation process.

For example: **C:\Program Files*My Company\My Project***

The Point SDK parses the public key token from the application and sends it to the Calyx servers. The servers validate the account and return either an approval or an error to the client.

The following information is sent to the Calyx servers for verification during activation:

- *Point Account ID* used for the Point application installation.
- Unique machine *GUID*.

- Public key token sent to Calyx for registration.

The activation process is invisible if the Point SDK is registered correctly. The activated token is stored on the client where the application is running after the activation is approved. An error message is received if the activation is not successful.

See Runtime Errors for additional information.

It is not possible to deactivate the token after the runtime activation is complete.

Point SDK Reference

The **PDK.DLL .NET assembly DLL** file is installed in the following Point application directory during the Point installation: C:\WINPOINT.

Add a reference to the file in the project before signing the assembly and using the Point SDK functions.

Adding a Point SDK (PDK.DLL) Reference

Use the following steps to reference the **PDK.DLL** file.

1. Open a new or existing project in **Microsoft Visual Studio**.
2. Right-click **References**.
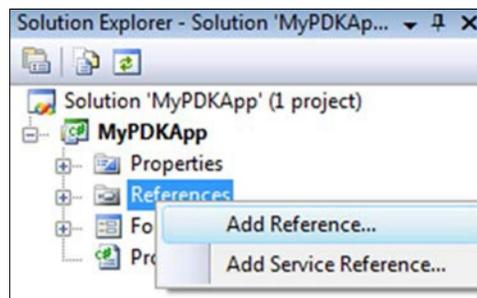
Figure 2: Solution Explorer Screen

3. Select **Add Reference**.

4. Select the **Browse** tab.
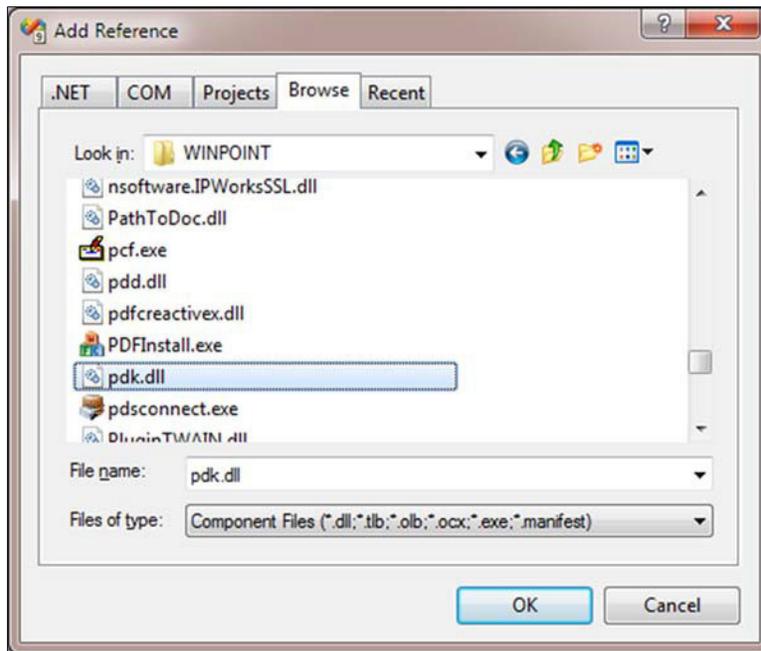
Figure 3: Add Reference

5. Navigate to the **PDK.DLL** file in **C:\WINPOINT**.
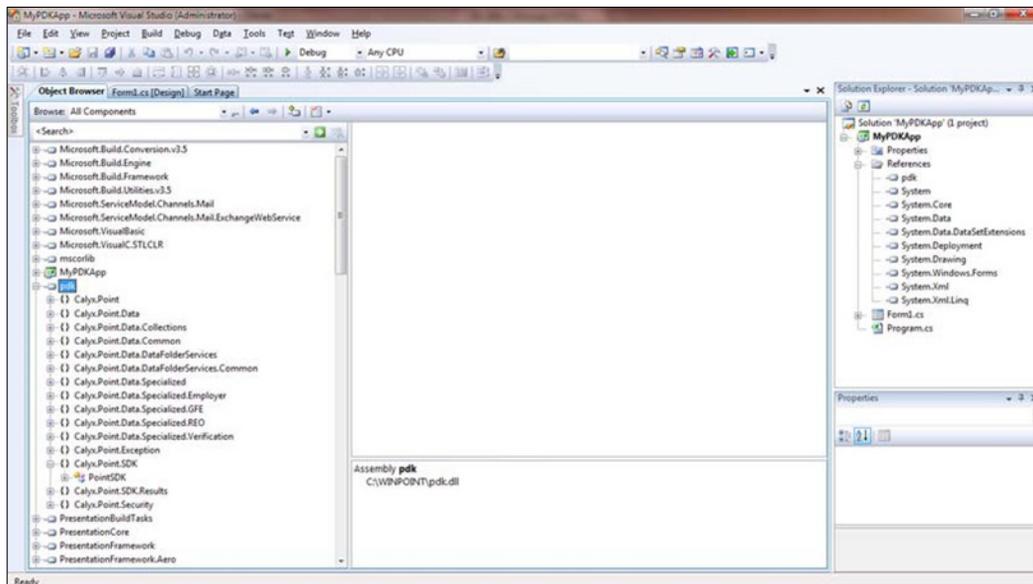
Figure 4: Visual Studio Administrator

6. Click **OK**.

7. Right-click **PDK**.

8. Select **Properties**.
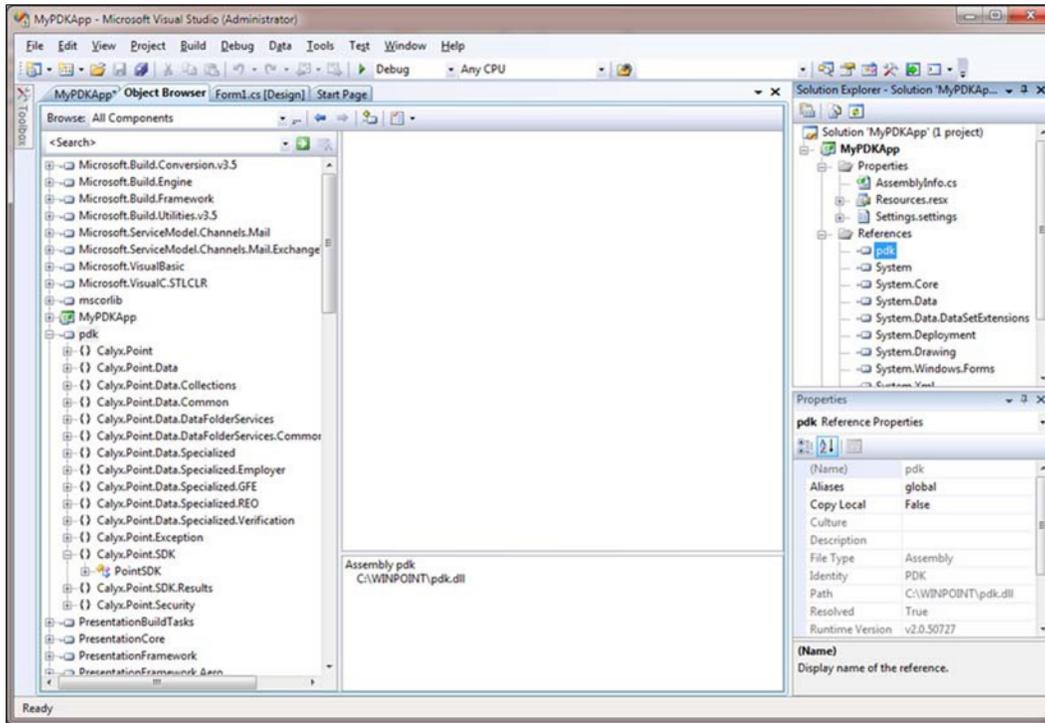
Figure 5: Verify Copy Local

9. Verify that **Copy Local** is set to **False**.

Assembly Signing

Calls to the Point SDK will not work if the assembly is not signed with a strong name that contains a public key. The public key must match the public key token that is embedded in the license file.

Sign Assembly Strong Name

Use the following steps to sign the assembly with a strong name.

1. Open the .NET project in Microsoft Visual Studio 2017.
2. Load the project properties from **My Project Properties**.
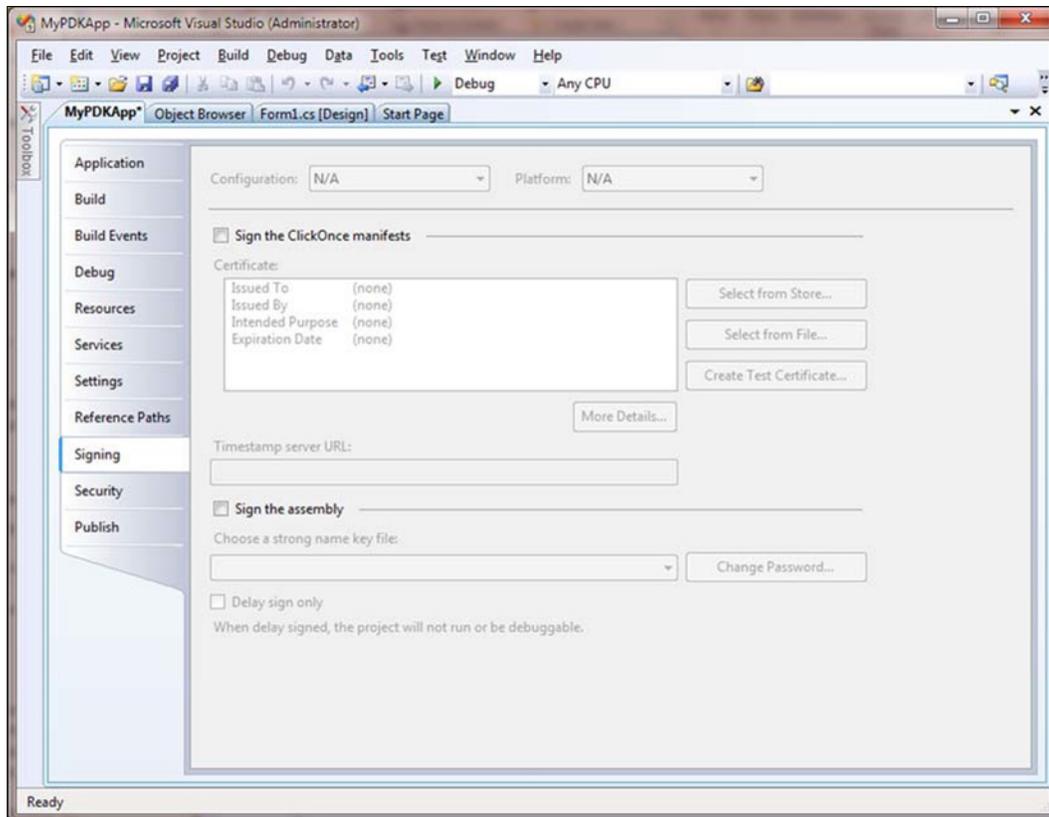
Figure 6: Properties Screen

3. Select **Signing** from the *MyProject* menu.
4. Select the **Sign the assembly** checkbox.
5. Select **Browse** from the dropdown list.
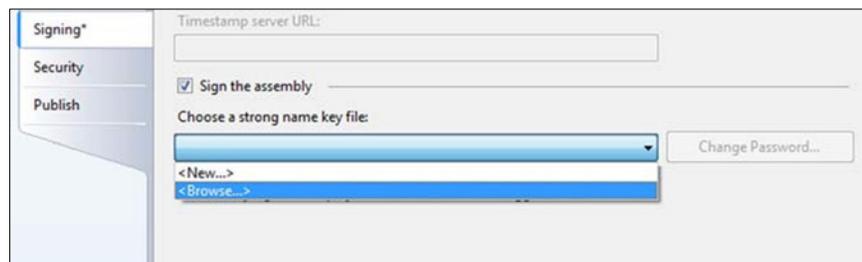
Figure 7: Browse Screen

6. Browse to the folder where the private key file is located.

Note: The private key file is in **C:\WINPOINT\sdk** if the private key is generated using the Calyx Software supplied batch file.

7. Select the private key file.

8. Click **Open**.

9. Select the **Private Key** file.
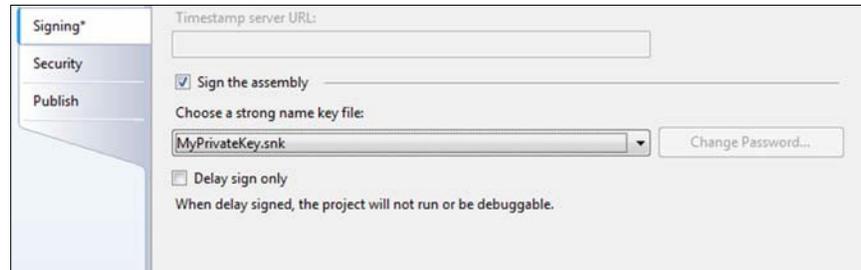
Figure 8: Private Key

10. Click **Open**.

11. Select **Save All** from the **File** menu to save the configuration.

Note: All future assemblies are signed with the strong name.

12. Select **Build** from the *Project* menu.

13. Select **Rebuild MyProject** to verify that the project builds without issues.

Note: The compiled assembly is stored in the *bin\debug* directory.

Verify Signature

Use the following steps to verify the assembly is signed with a strong name.

1. Open a **Microsoft Visual Studio 2017** command prompt.
2. Run the **.NET Strong Name** tool, *sn.exe*.

Figure 9: .NET Strong Name Tool

3. Verify the application is signed with a strong name and contains a public key.

Point SDK Methods

The Point SDK provides methods for accessing Point loan data and generating reports. The Point SDK honors security restrictions that are imposed in Point Administration and PointCentral. The Point user credential is passed to the login functions for further access to loans and report templates. Access rights are granted to the SDK object in the same way as with Point.

The following methods are the primary methods available in the Point SDK.

Table 7: Available Methods

Methods	Description
PointSDK	Point SDK accesses the Point SDK servers for activation when Point SKD is initiated for the first time
	The activation token is stored on the machine
ClientLogin	ClientLogin is the first step of interface.
	ClientLogin accepts the user ID and password to establish the secured environment to access data folders for loans.
	Loan and template accessibility is determined by the user login.
	Security behavior is the same as with the Point application.
GetVersionInfo	Retrieves the version information for Point and the Point SDK.
DataFolders	Retrieves the list of data folders that are accessible under the current login.
GetLoans	Retrieves a list of loans by search parameters that are passed.
	Search parameters are the same parameters used in Point Advanced Search.
Open	Opens a loan and creates an instance of the loan.
	Loans can be opened in read-only or read-write mode based on the passed parameters.
Create	Creates a new instance of a loan
GetData	Retrieves field data from the instance of the loan.
SetData	Updates field data to the instance of the loan.

Methods	Description
Save	Saves the data from the instance that is opened from Open.
SaveAs	Saves the data from the instance that is opened from Open or is created by Create.
Close	Close the opened loan instance.
CopyTo	Copies loan files from data folders to multiple data folders.
Delete	Deletes loan files from data folders.
GetReports	Creates a data set by using a predefined report template.

Class Definition

The Point SDK provides several classes used to develop applications that access Point data. The following diagram shows the typical order of the classes to be used in the code.
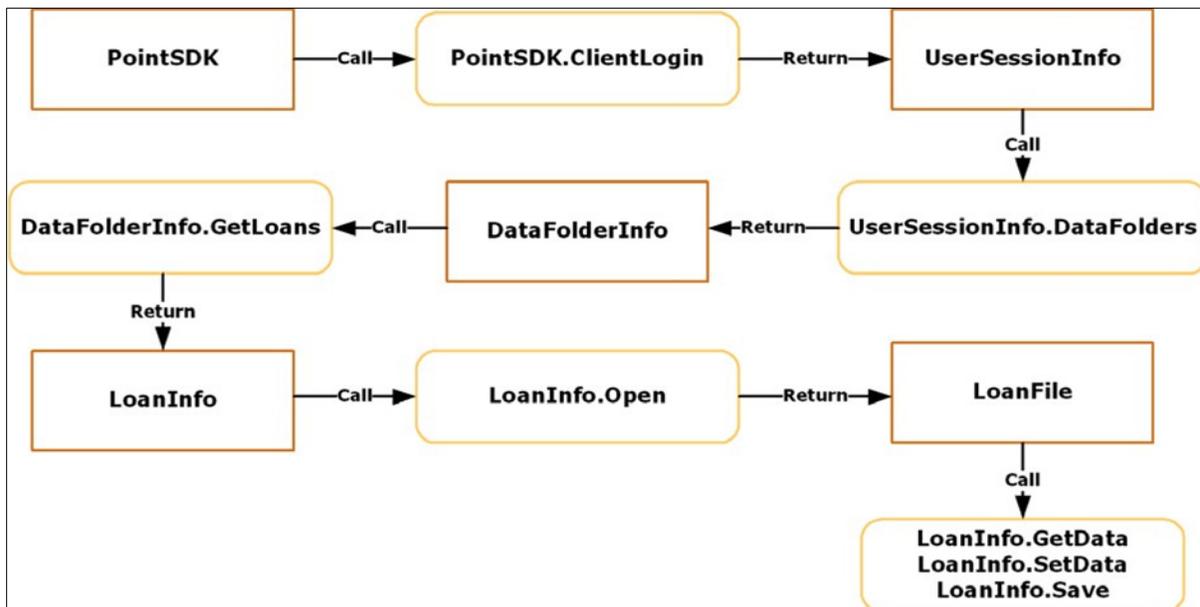
Figure 10: Class Diagram

Error Messages

Error messages can occur during the activation process on the development system or during runtime.

Refer to the message explanations for the cause of the error and the probable solution. Call the Calyx Software support number provided in the message for assistance when unable to resolve the issue.

Runtime Errors

Errors that can occur during runtime activation are described in the following table. Display the application error using the error information that corresponds to the error code. The error code can be caught through the exception class.

Table 8: Errors

Error code	Error message	Solution
-25	This software is not licensed for use with this Point Account ID. Please contact your software vendor for assistance.	A Point SDK-enabled account with a matching public key token and Point Account ID was found. The account is for a Point User SDK license and the SDK Account ID did not match the Point Account ID. Purchase a Vendor SDK license or ensure the correct Point Account ID is being used.
-24	Account ID for this software does not match the licensed Account ID. Please contact your software vendor for assistance.	A Point SDK-enabled account with a matching public key token was found, but the Point Account ID does not match the licensed Point Account ID.
-6	Internal Error	A non-Calyx client is attempting to connect to the Calyx servers.

Error code	Error message	Solution
-5	The request has timed out.	Verify that the date, time, and time zone are set correctly on the computer where the Point SDK is running.
-4	Internal Error	A client is attempting to submit an event, but is not using Secure Socket Layers (SSL).
-3	Internal Error	A non-Calyx client is attempting to connect to the Calyx servers.
-2	Internal Error	A non-Calyx client is attempting to connect to the Calyx servers.
-1	For PDK events: the sent SDK Account ID is blank.	A non-Calyx client is attempting to connect to the Calyx servers.
0	Success: found Point SDK enabled TMS account with matching Public Key Token and Account ID.	The activation was successful.
1	Internal Error	One of the web servers was unable to connect to the Calyx SQL database.
2	Internal Error	An unhandled exception occurred on one of the Calyx web servers.
3	Calyx is upgrading its servers.	Calyx is in the process of upgrading the servers. Try again in 15 minutes.