GeForce NOW SDK: Native Runtime

API Reference

SDK-GFN-003_v0.1 | 05 2019

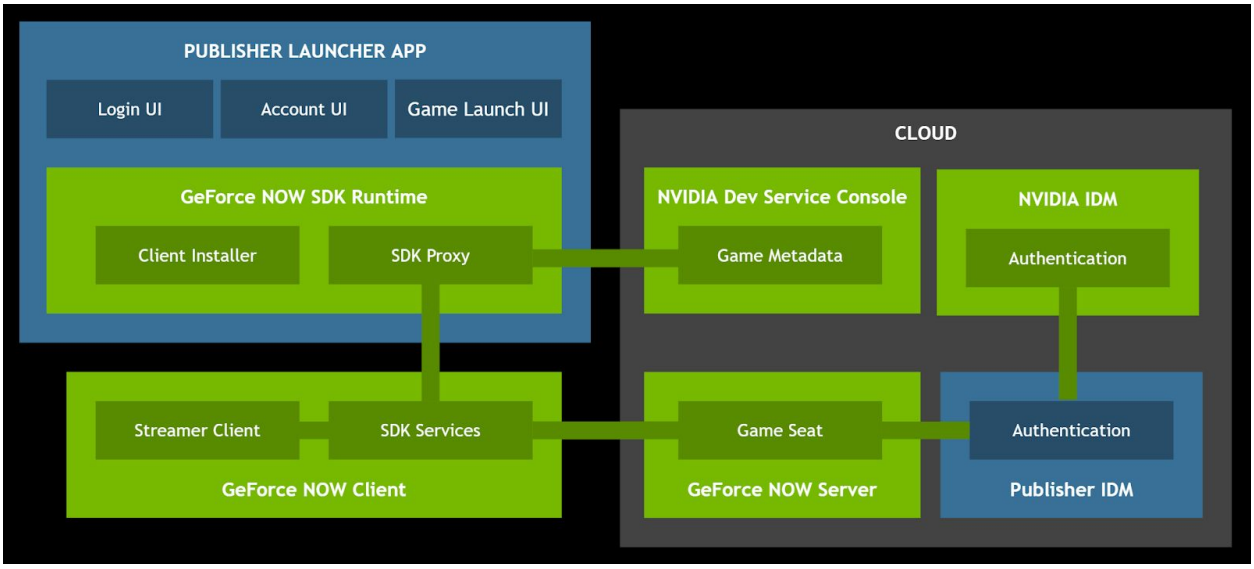
Document History

SDK-GFN-002_v1.0

Version	Date	Description of Change
0.1	*05/29/2019*	*Initial version*

Introduction

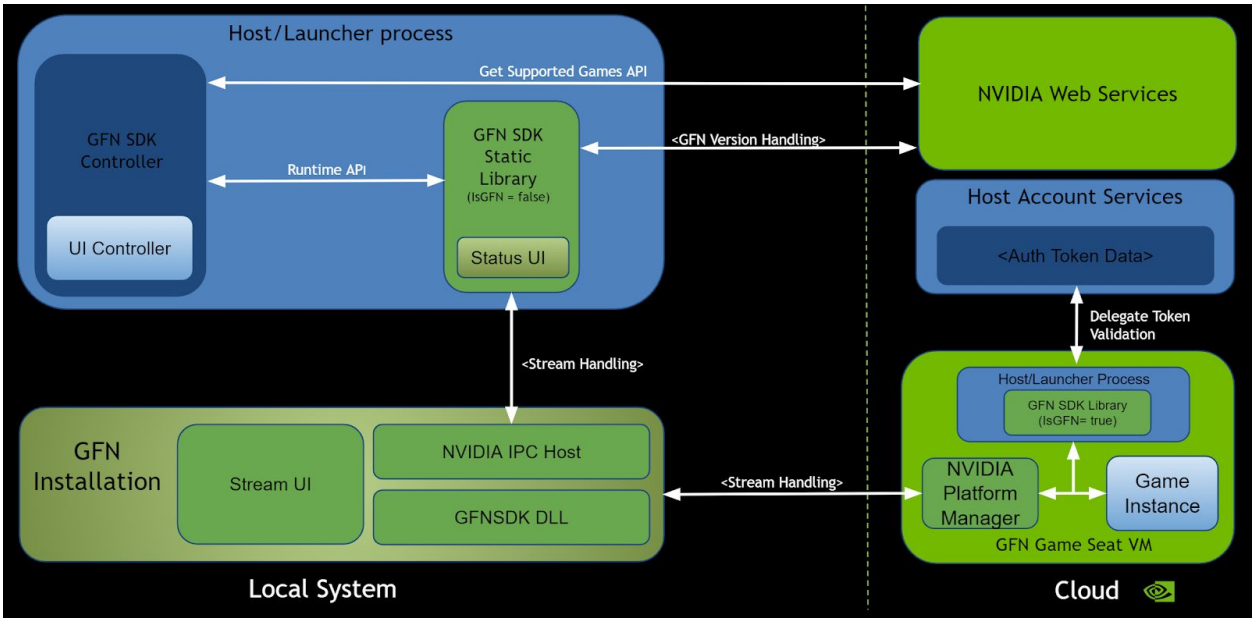
The NVIDIA GFN Runtime SDK provides a set of interfaces to allow game developers and game publishers to interact with parts of the NVIDIA GeForce NOW ecosystem. Integration is provided by various means, from native C interfaces to RESTful Web API calls, depending on the feature of the SDK.



This document provides details of how to integrate the native APIs of GFN Runtime SDK features into your application and its developer and deployment processes. For more detailed technical information about the architecture, refer to the architecture section later in this document.

Please note that the APIs described in this document are initial versions, and are subject to change in terms of parameters and consolidation in a near-future release.

Overview



The GFN Runtime SDK provides a static C library that is linked to the game/application and exposes the various APIs defined in this document. The behavior of those APIs depends on the environment the application is running in; either on a client/user system or in the GeForce NOW (GFN) cloud environment.

On client systems, this library checks for the presence of the GeForce NOW (GFN) client installation at initialization time. If the client is not present, or the client is out of date, then the library will initiate a download and installation of the latest client on the first API call that requires the client to be present, presenting UI to the user for the installation process.

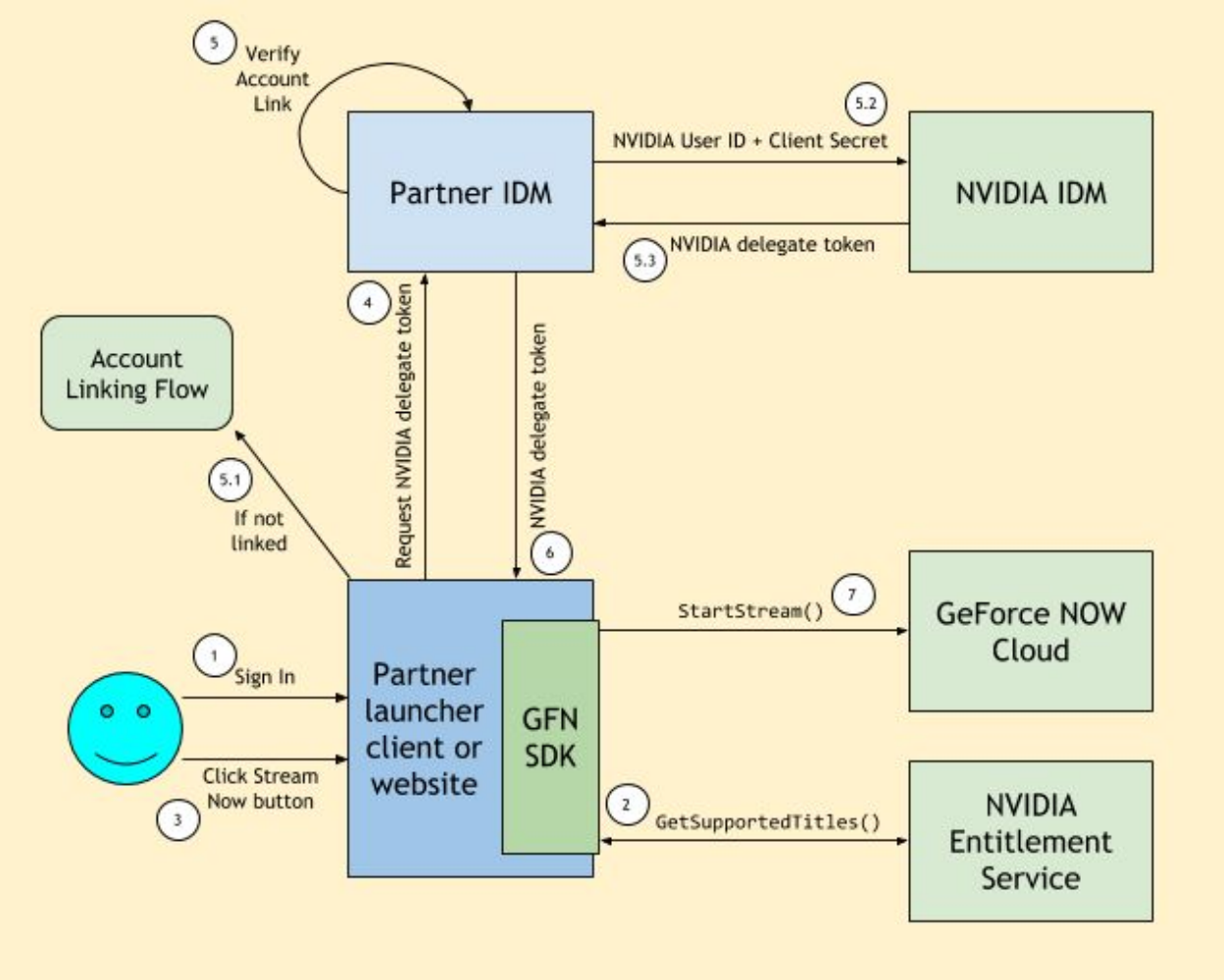
If the GFN client is installed on the system, the static library will defer API actions to components in the installation.

This design allows the static library to stay as thin as possible in the application, and provides backward and forward compatibility to new GFN client packages.

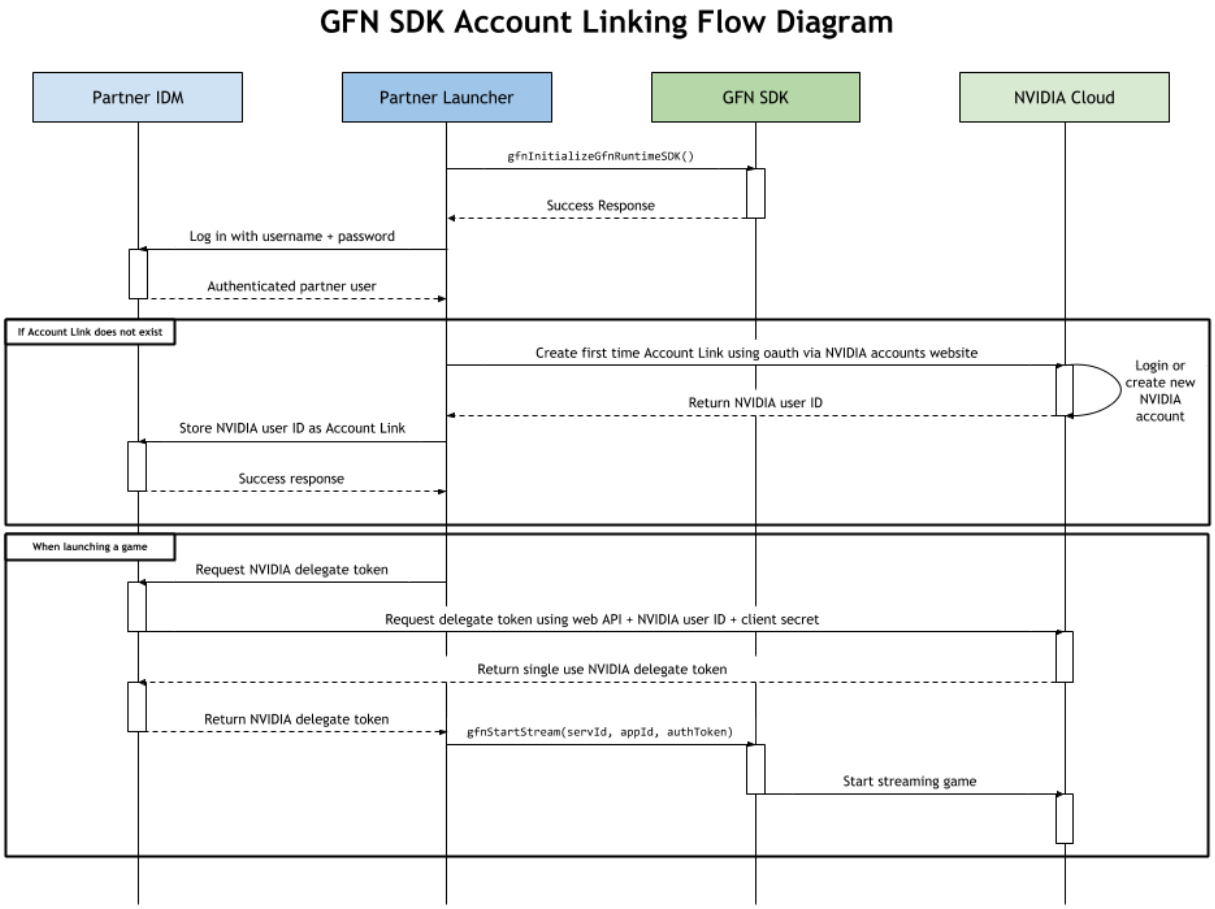
On GFN game systems, many of the APIs are no-ops as they apply only to client/end-user systems. In those cases, API calls will return a well-defined error code to denote the call was not applicable to the environment.

NVIDIA provides a backend IDM Service that application developers can use to validate users and obtain user information from. This provides a seamless flow for users to bypass multiple login steps in streaming a game through GFN.

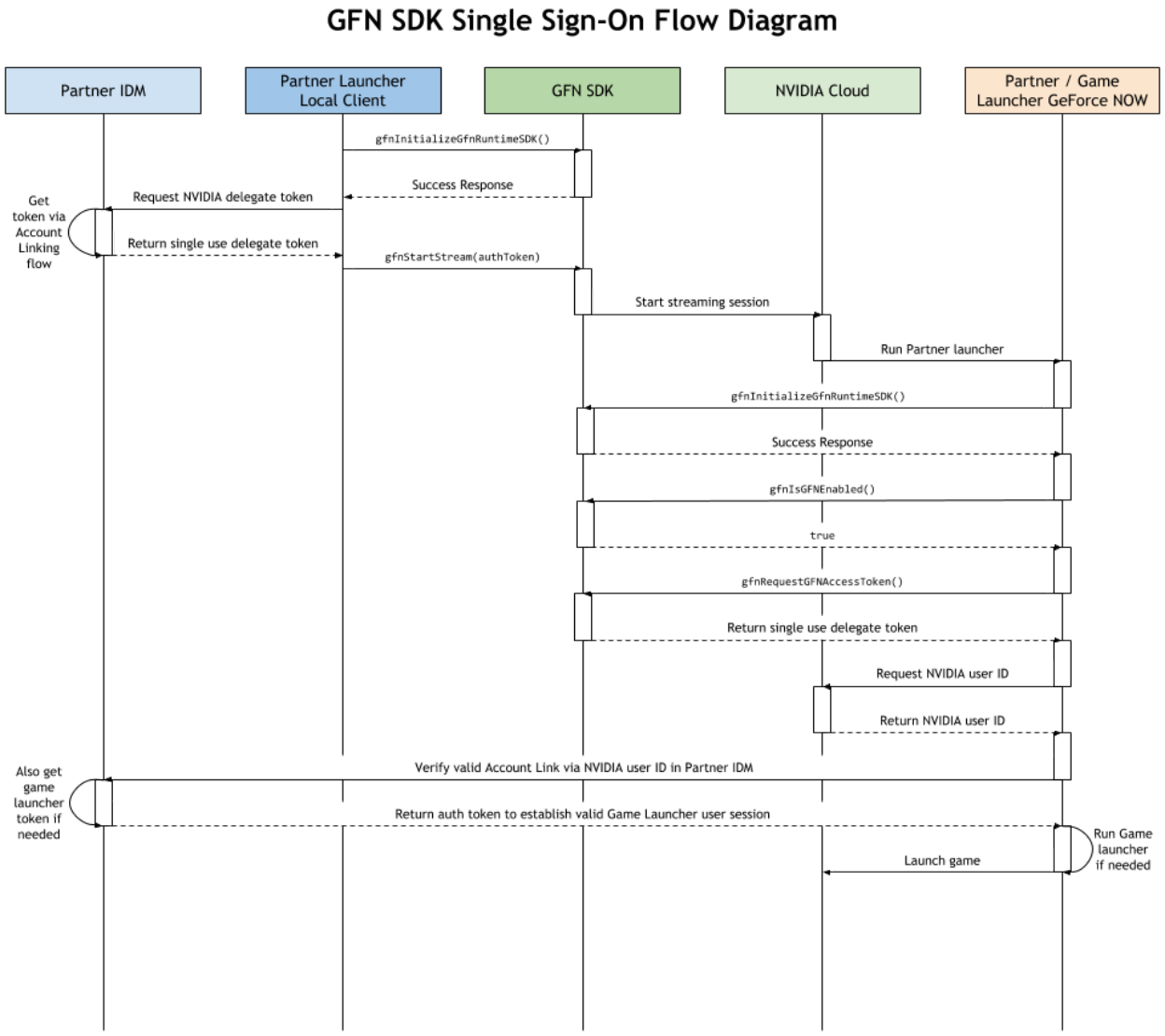
A simplified flow diagram of how this functions is shown below:



Authentication between the Partner IDM and NVIDIA IDM happens using secure HTTPS web API calls, and the account linking flow utilizes standard OAauth2 protocol. Once the account link is established the authentication process between Partner and NVIDIA becomes transparent to the user, and gaming streaming can be initiated without requiring any further authentication or manual login.



After the account link between Partner and NVIDIA has been established, that link can be utilized on the GFN server to facilitate Single Sign-On (SSO) so that the user does not have to manually login again, but all authentication happens transparently and the game launches immediately.



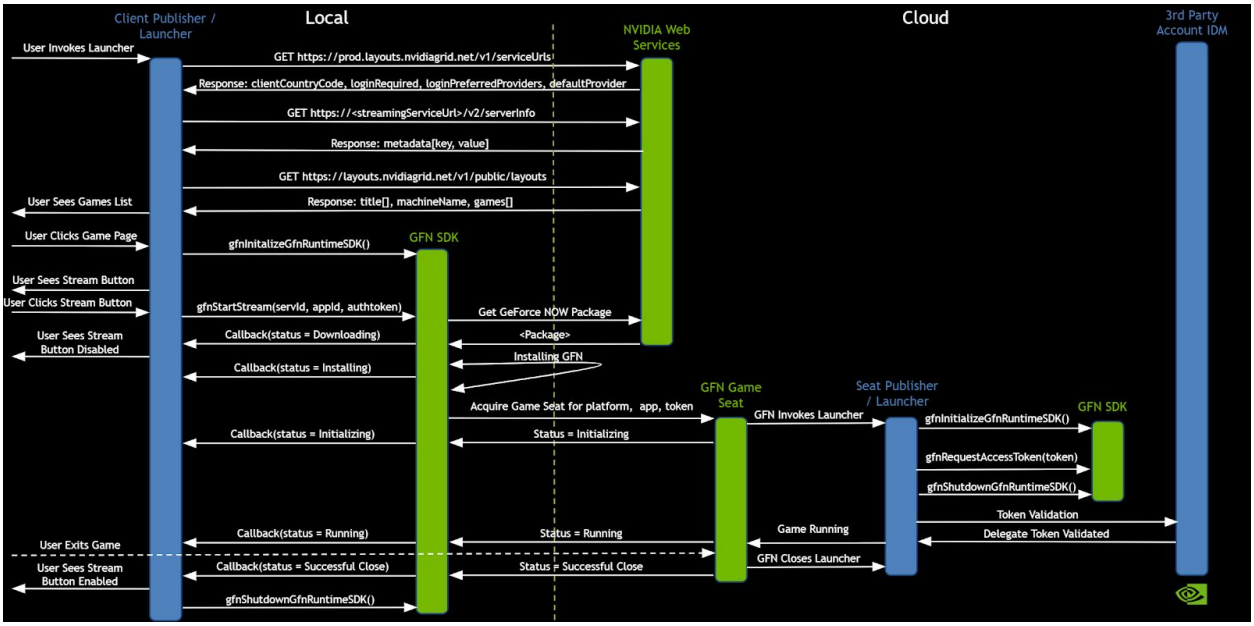
For additional high-level overview, please refer to the SDK primer available as part of the documentation section of the SDK's repository.

Key Concepts

GFN Runtime API methods are used to make requests from or to notify the GFN backend.

When your application is operating outside of the GFN environment, these methods are simple stubs that incur almost no cost, so it's safe to add these to your main build.

The calling convention differs by which API you've selected to use. In most cases, the methods return a GfnRuntimeError result, which can be used by the application to check for errors. In addition, some methods are asynchronous by nature, but provide synchronous variants when possible.



API Reference

Language	Definition
C	GfnRuntimeError **gfnInitializeGfnRuntimeSDK**()

Description

Should be called at application startup and prior to any GFN Runtime API methods.

When called from client environment (outside of a GFN cloud environment) it is expected for this method to always return success. In this case all GFN methods become no-ops and have no performance impact on your application.

Environment

Cloud and Client

Usage

Call as soon as possible during application startup.

Return values

When running in GFN environment:

GfnRuntimeError::gfnSuccess On success

GfnRuntimeError::gfnDLLNotPresent when supporting DLL not present (bad system configuration).

GfnRuntimeError::gfnIncompatibleVersion when GFN Runtime SDK library is not compatible with the supprting DLLs (bad system configuration).

Language	Definition
C	void **gfnShutdownGfnRuntimeSDK**()

Description

Releases the SDK, and frees up memory allocated by GFN and disconnects from GFN backend.

Environment

Cloud and Client

Usage

Call during application shutdown or when GFN Runtime API methods are no longer needed.

Language	Definition
C	GfnRuntimeError **gfnGetClientIp**(const char** ppchClientIp)

Description

Gets user's client IPv4 address

Since Applications running under GFN run in the GFN data centers, any IP queries made by the Application will return IPs associated with the data center, not the user's client IP. This allows the Application obtain the client IP in v4 format so that developers can make regional business decisions.

Environment

Cloud

Usage

Call this during application start or from the platform client in order to get the user's actual client IP address.

Parameters

ppchClientIp: Output IPv4 in string format. Example: "192.168.0.1".

Return value

gfnSuccess: On success

Otherwise, appropriate error code.

Language	Definition
C	GfnRuntimeError **gfnGetClientLanguageCode**(const char** ppchLanguageCode)

Description

Gets user's client language code in the form <lang>-<country> using the standard ISO 639-1 language codes and ISO 3166-1 Alpha-2 country codes.

Environment

Cloud

Usage

Call this during application start or from the platform client in order to get the user's language and country settings.

Parameters

ppchLanguageCode: Language code as a string (for example, "en-US").

Return value

gfnSuccess: On success

Otherwise, appropriate error code

Language	Definition
C	bool **gfnIsRunningInCloud**()

Description

Determines if calling application is running in GFN cloud environment or not.

Environment

Cloud and Client

Usage

Use to determine whether to enable / disable any GFN cloud environment specific application logic, for example, to block any calls to gfnStartStream(), or calls to gfnRequestKeyboardOverlayOpen() for input if the function returns true.

Return value

true: Application is running on a game seat virtual machine (cloud environment) or GFN test environment.

false: Application is not running in a GFN cloud environment.

Language	Definition
C	GfnRuntimeError **gfnRequestGfnAccessToken**(const char** ppchToken)

Description

Request to obtain a user specific access token to allow access to the GFN backend service (IDM endpoint).

Environment

Cloud

Usage

The access token provided can be used by the application's backend servers to validate the user and obtain user data from the GFN backend service. The GFN backend service provides an OAuth2 interface for validating users and retrieving data. See Account Federation section for more information.

Parameters

ppchToken: Populated with a user specific GFN access token.

Return value

gfnSuccess: On success

Otherwise, appropriate error code.

Language	Definition
C	GfnRuntimeError **gfnStartStream**(char* pchServiceId, char* pchAppId, char* pchAuthToken, StartStreamCallbackSig cb, void* context, int timeoutMs);

Description

Requests GFN client to start a streamed session of an application.

Environment

Client

Usage

Use to start a streaming session.

Parameters

pchServiceId: Identifier of the launcher service, e.g. "Steam".

pchAppId: Identifier of the requested application to start streaming.

pchAuthToken: Identifier of the requested application to start streaming.

cb: Pointer to StartStreamCallbackSig callback function to be called with streaming events.

context: User context for the callback.

timeoutMs: Time after which attempt to start streaming will be aborted.

Return value

gfnSuccess: On success

Otherwise, appropriate error code

Language	Definition
C	GfnRuntimeError **gfnRegisterExitCallback**(ExitCallbackSig exitCallback, void *pUserContext)

Description

Register an application callback with GFN to be called when GFN needs to exit the game.

Callback function signature is:

```
GfnApplicationCallbackResult func(void* pUserContext)
```

Where the pUserContext pointer is the unmodified user context pointer provided by the caller as an argument to the registration method (can be NULL).

Game application should save the game and exit when this callback is called.

Environment

Cloud

Usage

Call to register an exit callback with GFN.

Parameters

exitCallback: Pointer to ExitCallbackSig callback function.

pUserContext: Pointer to user defined context object, which will be returned as a parameter to the callback (can be NULL).

Return value

gfnSuccess: On success

Otherwise, appropriate error code.

Language	Definition
C	GfnRuntimeError **gfnRegisterPauseCallback**(PauseCallbackSig pauseCallback, void *pUserContext)

Description

Register an application callback with GFN to be called when GFN needs to pause the game on the user's behalf.

Callback function signature is:

```
GfnApplicationCallbackResult func(void* pUserContext)
```

Where the pUserContext pointer is the unmodified user context pointer provided by the caller as an argument to the registration method (can be NULL).

Game application should save and pause the game when this callback is called. For Multiplayer games, it is recommended that this is implemented similar to a client disconnect.

Environment

Cloud

Usage

Call to register a pause callback with GFN

Parameters

pauseCallback: Pointer pauseCallback to callback function.

pUserContext: Pointer to user defined context object, which will be returned as a parameter to the callback (can be NULL).

Return value

gfnSuccess: On success

Otherwise, appropriate error code.

Language	Definition
C	GfnRuntimeError **gfnRegisterInstallCallback**(InstallCallbackSig installCallback, void *pUserContext)

Description

Register an application callback with GFN to be called when GFN needs to complete an application installation.

Callback function signature is:

GfnApplicationCallbackResult func(const TitleInstallationInformation* pInfo, void* pUserContext)

Where pInfo is a pointer to a TitleInstallationInformation struct containing:

- pchPlatformAppId: The ID of the title that needs to be installed
- pchBuildPath: The build path where the title binaries have been attached
- pchMetadataPath: The path to any optional metadata for the title

pUserContext pointer points to the user context pointer provided by the caller as an argument to the registration method (can be NULL).

The Application should use the provided paths to complete the install of a title and make it ready

to launch.

Environment

Cloud

Usage

Call to register an install callback with GFN

Parameters

installCallback: Pointer to InstallCallbackSig callback function.

pUserContext: Pointer to user defined context object, which will be returned as a parameter to the callback (can be NULL).

Return value

gfnSuccess: On success

Otherwise, appropriate error code.

Language	Definition
C	GfnRuntimeError **gfnRequestKeyboardOverlayOpen**(GfnScreenPosition gspPosition)

Description

Called from application when it is expecting text input from user. Calling this API would trigger a native keyboard overlay to be shown to the GFN user such that he/she can most easily enter text based on the particular GFN client platform being used.

There's no special input handling needed from application; input will be injected into application by GFN (as is done in all other times running in GFN). Note GFN is not displaying any text input box or prompt to user, only a keyboard overlay.

Environment

Cloud

Usage

This API should be called as a pair with RequestKeyboardOverlayClose. Multiple calls to RequestKeyboardOverlayOpen will have no effect after the first call.

Parameters

gspPosition: the desired screen positioning of text input element (i.e. Android keyboard). Should be one of the following values:

- gspBottom
- gspTop
- gspLeft
- gspRight
- gspCenter
- gspTopLeft
- gspTopRight
- gspBottomLeft
- gspBottomRight

Return value

gfnSuccess: On success

Otherwise, appropriate error code.

Language	Definition
C	GfnRuntimeError **gfnRequestKeyboardOverlayClose**()

Description

Called from application when necessary text input has been processed and user can continue. This would cause a previously requested keyboard overlay on the GFN user's client display to be dismissed**.**

Environment

Cloud

Usage

RequestKeyboardOverlayOpen should be called before this method. If not, RequestKeyboardOverlayClose will have no effect.

Return value

gfnSuccess: On success

Otherwise, appropriate error code.

Language	Definition
C	GfnRuntimeError **gfnSetupTitle**(const char* pchPlatformAppId)

Description

Notifies GFN that an application should be readied for launch. GFN will ready the application by attaching its virtual drive, downloading any game specific metadata and downloading user save data.

Will block until application setup is complete and the application is ready for install. If a callback has been registered via RegisterInstallCallback, it will then be called. If no install callback is registered, GFN will attempt to perform the install instead.

Environment

Cloud

Usage

Use prior to installing an application.

Parameters

pchPlatformAppId: Identifier of the requested application to prepare.

Return value

gfnSuccess: On success

Otherwise, appropriate error code.

Language	Definition
C	GfnRuntimeError **gfnTitleExited**(char* pchPlatformId, const char* pchPlatformAppId)

Description

Notifies GFN that an application has exited. GFN will then start the shutdown process for that application.

Note that this is for use by platform clients only and assumes the application has already completed execution. To shutdown from within an application itself, ShutdownGFNLinkSDK is used.

Environment

Cloud

Usage

Use after an application has exited.

Parameters

pchPlatformId: Identifier of the launcher service, e.g. “Steam”.

pchPlatformAppId: Identifier of the requested application to start streaming.

Return value

gfnSuccess: On success

Otherwise, appropriate error code.