GO-GLOBAL

Application Programming Interfaces

Version 4.8.2

COPYRIGHT AND TRADEMARK NOTICE

CONTENTS

1 INTRODUCTION

GO-Global provides remote access to Windows applications that are installed on centrally-managed servers. Applications do not have to be modified to be accessed via GO-Global, but there are times when it is advantageous to customize an application for execution in a multi-user, hosted environment. For example, if an application takes a long time to start, developers can provide quicker access to the application by maintaining a cache of pre-loaded application instances.

Alternatively, developers may wish to embed a hosted application within another application on the client device and send custom messages between the hosted application and the client application. To support these and other usage scenarios, GO-Global provides the following Application Program Interfaces (APIs).

SESSION MANAGER API

The Session Manager API is a Windows, C language interface to the GO-Global host. It supports development of Windows applications that manage and monitor clusters of GO-Global hosts. The Session Manager API uses Integrated Windows Authentication to authenticate with the host. Therefore, processes that call Session Manager APIs must be run under user accounts that are defined in the domain to which the host belongs, or in a domain that is trusted by the host's domain. Most Session Manager APIs can only be called from an account that is a member of the host's Administrators group.

The Session Manager API includes an interface to the ActiveX Client that allows the GO-Global client to be embedded within Windows applications.

The Session Manager API is supported on Windows.

GATEWAY API

The Gateway API is a RESTful interface to the GO-Global gateway. It supports development of both end-user applications and custom session management applications. It enables seamless integration of desktop applications and web applications. It is supported on Windows, Linux, Mac OS, iOS, and Android, and it works with any programming language.

BROWSER COMPONENT API

When an application running on a GO-Global host is accessed via a GO-Global Add-on running in a browser, the Browser Component API allows the application to call JavaScript functions in the browser on the client. The Browser Component forwards calls made from the application on the host to specified JavaScript functions in the browser on the client via the connection between the GO-Global host and the GO-Global Add-on. The Browser Component API is supported on Windows clients by the GO-Global Add-ons for Internet Explorer, Mozilla Firefox and Google Chrome, and it is supported on SUSE Linux clients by the GO-Global Add-on for Mozilla Firefox.

CLIENT PROCESS MANAGER API

The Client Process Manager API allows applications running on GO-Global hosts to start and stop processes on the client computer. The Client Process Manager API is supported on Windows clients.

COMPONENT API

The Component API is a general purpose interface that lets developers extend the GO-Global client and send custom messages between an application running on a GO-Global host and a custom DLL running in the GO-Global client. The Component API is supported on Windows and Linux clients.

In summary, the Session Manager and Gateway APIs allow developers to create applications that manage GO-Global sessions, and the Browser Component, Client Process Manager and Component APIs allow applications running in GO-Global sessions to communicate with the client. This document provides an overview of these APIs and illustrates how they can be used to integrate GO-Global with other applications and web services.

2 SESSION MANAGER API

The Session Manager API is a C language interface to the GO-Global host. It exposes the functions that the GO-Global Cluster Manager uses to manage and monitor GO-Global hosts. Specifically, it provides functions that perform the following operations:

- Connect and authenticate to a GO-Global host
- Publish applications
- Create and terminate sessions
- Start and stop processes
- Authenticate clients
- Disconnect clients
- Enumerate running sessions and processes
- Receive session and connection event notifications.

The Session Manager API uses Integrated Windows Authentication to authenticate with hosts. Therefore, processes that call Session Manager APIs must be run under user accounts that are defined in the domain to which the host belongs, or in a domain that is trusted by the host's domain. In addition, most Session Manager APIs require that the calling process be running under an account that is a member of the host's Administrators group. Because of this, most Session Manager APIs are not suitable to be called from end-user applications.

For end user applications, GraphOn provides an ActiveX Client and browser Add-ons for Mozilla Firefox, Apple Safari, and Google Chrome. The ActiveX Client can be run in Internet Explorer through a standard browser interface, or it can be embedded in a Windows application using GraphOn's ActiveX Client API.

Developers can use the above clients in the frontend of a third-party application and use the Session Manager API in the application's backend. For example, *Figure 1* illustrates how the backend of a third-party application can use the Session Manager API to pre-launch an application in a GO-Global session and then display the hosted application to a user in the frontend of the application using the ActiveX Client.

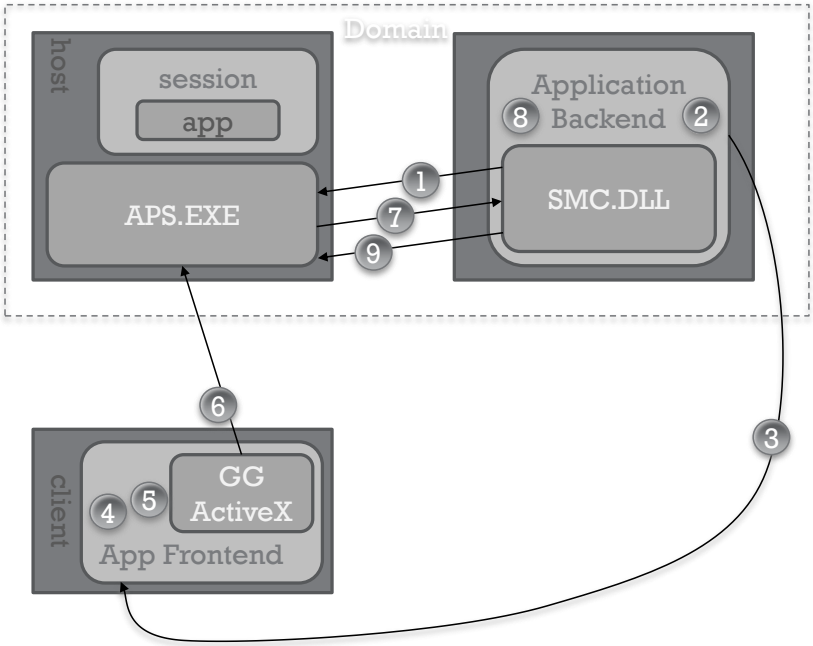

Figure 1: Start Session and Connect Client with Session Manager API

The steps to this process are as follows:

1. Start a session and launch an application on the host via the Session Manager API:
    - Call **SMI_Initialize** to open a connection from the application backend to APS.EXE on the host.
    - Call **SMI_CreateSession** to create a session.
    - Call **SMI_LaunchApplication** to start an application.
2. Generate one-time use credentials, generally a unique string.
    - The credentials are application-defined; GO-Global will store them in the **SMI_SESSION_CREDENTIALS** structure, but it will not validate them.
3. Transmit host address, session ID, and client credentials to the application's frontend.
    - GO-Global does not provide a way to transmit this data from the application's backend to its frontend.
    - The transmission must be secure (e.g., via HTTPS).
4. Load the ActiveX Client and set its host address, session ID, and client credentials parameters.
    - Call the ActiveX Client's **put HostName**, **put Session**, **put SessionCredentialsString** and **put SesssionCredentialsAuthority** functions.
5. Connect the client to the host.
    - Call the ActiveX Client's **Connect** function.

6. ActiveX Client connects to host and sends its session ID and credentials to host.
    - Client credentials are encrypted using RSA algorithm and 512-bit public key.
    - Configure host to use SSL to prevent man-in-the-middle attack.
7. APS.EXE sends credentials to application backend via SMC.DLL.
    - SMC.DLL calls Session Manager Interface's **onClientConnect** callback function.
8. Application backend authenticates client using credentials.
    - Application backend verifies that the credentials match those generated in Step 2.
9. Application backend connects the client to the session.
    - Application backend calls **SMI_ConnectClient**.
    - APS.EXE connects the client to the session.
    - Application that was started in Step 1 is displayed to the user.

The Session Manger API is support on Windows.

3 GATEWAY API

The Gateway API is a RESTful web interface to the GO-Global gateway. Specifically, it provides functions that perform the following operations:

- Authenticate with a GO-Global gateway
- Publish applications
- Create and terminate sessions
- Start and stop processes
- Create and terminate private workspaces
- Authenticate clients
- Disconnect clients
- Enumerate sessions and processes

This functionality is very similar to the functionality provided by the Session Manager API. The main differences between the Gateway API and the Session Manager API are:

- As the name implies, the Gateway API requires the GO-Global gateway; the Session Manager API does not.
- The Gateway API is a web-based interface; the Session Manager API is a Windows interface.
- The Gateway API is supported on Windows, Linux, Java, Mac OS, iOS and Android; the Session Manager API is supported only on Windows.
- The Gateway API supports virtually any programming language; the Session Manager API supports C.
- The Gateway API does not provide session and connection event notifications; the Session Manager API does.
- Since the Gateway API interfaces with the GO-Global gateway, it supports larger scale deployments than the Session Manager API.
- The process of connecting clients to sessions is much easier with the Gateway API than it is with the Session Manager API.

To illustrate this last difference, *Figure 2* shows how to connect a client to a session using the Gateway API.

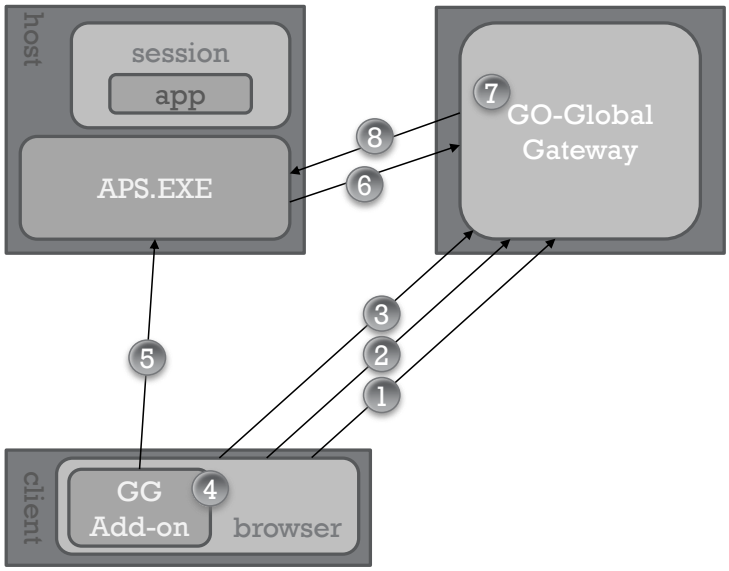

Figure 2: Start Session and Connect Client using Gateway API

Although the overall number of steps is approximately the same as the number of steps needed to perform the same operation with the Session Manager API, most of the steps are performed by GO-Global. Only steps 1-4 need to be performed by the third-party application.

1. Authenticate with the gateway
    - Send **getPortalUser** request
2. Create a session and start an application
    - Send **createSession** and **openFile** requests
3. Request a URL that can be used to connect a client to the session
    - Send c**onnectClient** request
4. Start GO-Global Add-on
    - Browser to URL

When the Add-on is loaded, GO-Global automatically performs the tasks necessary to connect the client to the session:

5. Connects to host and sends session ID and credentials to host.
    - Client credentials are encrypted using RSA algorithm and 512-bit public key.
    - Configure host to use SSL to prevent man-in-the-middle attack.
6. APS.EXE sends credentials to gateway.
7. Gateway authenticates client using credentials.
    - Gateway verifies that the credentials match those generated in Step 3.
8. Gateway connects the client to the session.
    - Application that was started in Step 2 is displayed to the user.

The previous example illustrates how the Gateway API being can be called directly from the frontend of a web application running in a browser. The Gateway API can also be called from an application's backend as shown in *Figure 3*. In this example, the application's backend manages a pool of pre-initialized sessions so an application can be presented to users very quickly when the user requests it.

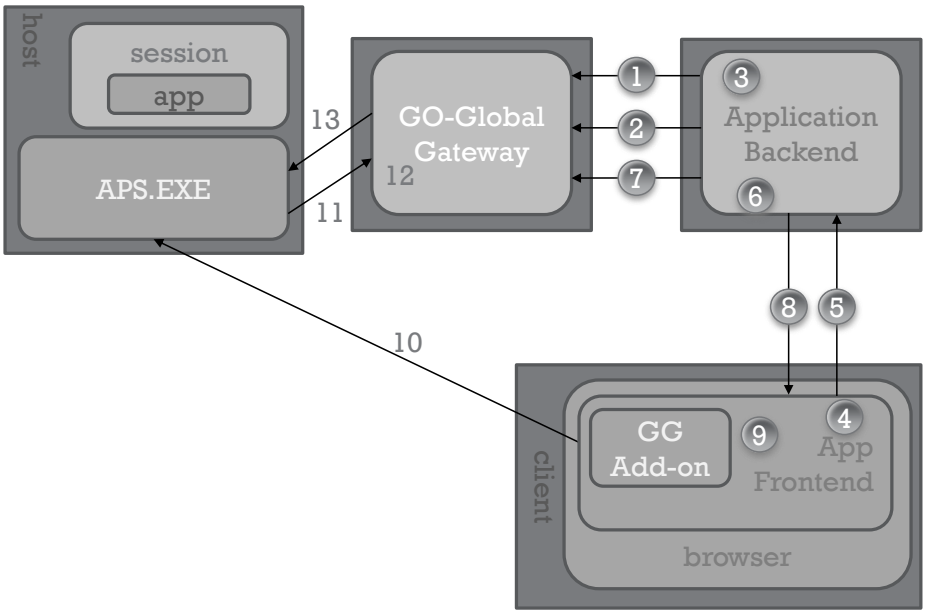

Figure 3: Session Pool Management using Gateway API from Application Backend

In this example, the steps are as follows:

1. Application backend authenticates with the gateway.
    - Sends **getPortalUser** request
2. Application backend creates and initializes sessions.
    - Sends **createSession** and **openFile** requests to pre- initialize sessions
3. Application backend adds sessions to its database of pre-initialized sessions.
4. User selects a hosted application via frontend's user interface.
5. Frontend sends request for hosted application to backend.
6. Backend checks a pre-initialized session out of its pool.
    - Finds a session with the desired application and removes it from the pool
7. Backend requests a URL for the session.
    - Sends **connectClient** request to the gateway
8. Backend returns URL to the frontend.
9. Frontend loads the GO-Global Add-on.

GO-Global then performs the tasks of connecting the client to the session:

10. Add-on sends connection credentials to APS.EXE.
11. APS.EXE sends the connection credentials to the gateway.
12. The gateway authenticates the connection requests.
13. The gateway connects the client to the session.

The Gateway API is supported on Windows, Linux, Mac OS, iOS and Android, and it works with any programming language.

4 BROWSER COMPONENT API

The Browser Component is a GO-Global object that allows applications running in GO-Global sessions to call JavaScript functions in web pages that embed a GO-Global Add-on. *Figure 4* illustrates how this is done.

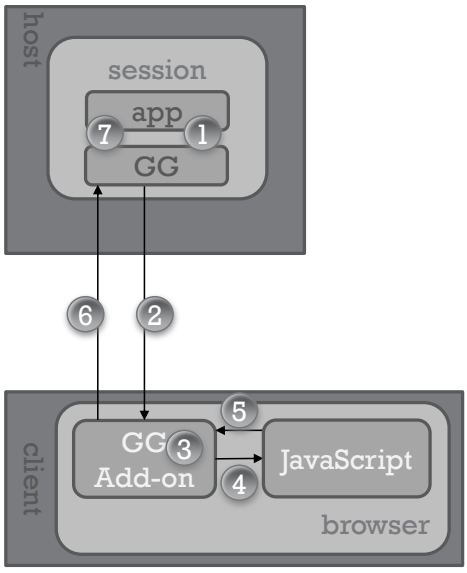

Figure 4: Calling JavaScript functions in a Browser from an Application Running in a GO-Global Session

The steps are as follows:

1. The application calls the Browser Component's **sendMessage** function.
2. The Browser Component sends the message to the GO-Global Add-on through the Add-on's connection to the host.
3. The Add-on queues the message to the thread in the browser that executes JavaScript.
4. The Add-on calls the specified JavaScript function from the thread that executes JavaScript.
5. The JavaScript function returns the result to the Add-on.
6. The Add-on returns the result to the GO-Global host.
7. The GO-Global host returns the result to the application.

The Browser Component API is supported on Windows clients by the GO-Global Add-ons for Internet Explorer, Mozilla Firefox, and Google Chrome.

5 CLIENT PROCESS MANAGER API

The Client Process Manager API allows applications running in GO-Global sessions to start and stop processes on the client computer. Specifically, it allows applications to:

- Start processes on the client computer.
- Wait for processes on the client to exit.
- Terminate client processes.
- Retrieve process exit codes.
- Retrieve error codes

The Client Process Manager commands are transmitted from the application running on the host to the GO-Global client through the client's connection to the host.

The Client Process Manager API is supported on Windows clients. The GO-Global Client Process Manager API [hyperlink] document provides a specification of the API's functions.

6 COMPONENT API

The Component API exposes the interface that the Browser Component and Client Process Manager APIs use to transmit messages between the client and host. It is a general purpose interface that lets developers extend the GO-Global client and send custom messages between a pair of custom libraries, one running in an application running on a GO-Global host, and the other running in the GO-Global client. It is generally used to allow applications running in GO-Global sessions to interface with custom hardware and applications on the client.

The Component API provides functions that:

- Register custom components with the GO-Global host and client.
- Instantiate instances of custom components.
- Obtain references channels that provide socket-like interfaces for data transmission.
- Send custom messages from the host to the client and vice versa.
- Release instances of custom components.

The Component API is supported on Windows and Linux clients. A specification of the API's function is available in the Component SDK [hyperlink] document.

7 ADDITIONAL INFORMATION

Detailed information about each of the above APIs is provided in the following documents:

- **GO-Global Session Manager API:** Documents the Session Manager API's functions and data structures.
- **GO-Global Session Manager API Samples:** Describes sample applications that demonstrate how to use the Session Manager API.
- **GO-Global Gateway API:** Documents the Gateway API's requests.
- **GO-Global Gateway API Sample:** Describes a sample application that demonstrates how to use the Gateway API.
- **GO-Global Browser Component API:** Specifies the Browser Component API's functions and describes how to use them.
- **GO-Global Client Process Manager API:** Specifies the Client Process Manager API's functions and describes how to use them.
- **GO-Global Component SDK:** Specifies the Component API's functions and data structures.