# External Programming and Authentication APIs

## Overview

In some cases, it is useful to create external web programs that can be used along with a Profound UI / Genie session. For example, external CGI programs can be written to provide options for drop down boxes, auto complete fields, or even to drive a Profound UI screen that is loaded into an IFrame. In most cases, it's desirable to run external CGI programs under the same user authority and with the same library list as the Profound UI / Genie session job.

This can be accomplished using a set of APIs that are packaged with Profound UI.

## Authority and Library List Synchronization APIs

Using the authority and library list synchronization APIs requires binding to a service program PUISYNCR in the Profound UI product installation library. A binding directory can be created with an entry for this service program. It's very important that external CGI programs activate the copy of the service program that resides in the Profound UI installation library where the Profound UI / Genie session is running from. If multiple copies of Profound UI are installed on the server and a CGI program activates a copy of PUISYNCR from the wrong installation library, the APIs will fail. The programs can ensure the correct copy is used by either using qualified service program binding, or by adjusting the library list. A copy member PUISYNCR in file QRPGLEINC is provided in the product installation library, which includes the API prototypes.

The APIs work by reading the Profound UI / Genie session id. The session id is a 64-character string, and is automatically passed on all external program calls that are initiated through Profound UI / Genie Designer widget properties. Such as 'chart url', 'choices, url', etc. The session id is passed as a query string parameter called 'AUTH'.

If external program calls are initiated in other ways (such as through a manual Ajax call, through an IFrame page, etc.), the session id must be passed manually to the external program. The session id is available to JavaScript in all Profound UI / Genie sessions through the following global property:

```
pui.appJob.auth
```

## APIs for RPGsp Pages

For RPGsp pages, the following APIs can be used. The APIs expect that a parameter called 'AUTH' is present on the page request. The APIs will read the session id from this parameter internally.

**SyncJob():** Sets the job current user and library list to match the related Profound UI / Genie session. An optional indicator parameter can be passed *Off to avoid setting the library list. The procedure returns an error message on failure, or empty string if successful.

**ResetJob():** Sets the job current user back to the standard CGI user profile. This MUST be called after any successful call to **SyncJob()**, or the job will be left in an invalid state and may fail on the next page request.

**RtvJobInfo():** Returns a Job Information data structure containing details about the Profound UI / Genie session job. See below for the DS definition. The procedure returns an error message on failure, or empty string if successful.

## APIs for Other Types of CGI Programs

For CGI programs written with frameworks other than RPGsp, the following APIs can be used. These APIs work by accepting the session id as a parameter.

**pui_SyncJob():** Sets the job current user and library list to match the Profound UI / Genie session, as indicated by the passed session id. Returns a job information data structure. An escape message is sent if there are any errors.

**pui_ResetJob():** Sets the job current user back to the standard CGI user profile. This MUST be called after any successful call to **pui_SyncJob()**, or the job will be left in an invalid state and may fail on the next page request.

**pui_RtvJobInfo():** Returns a Job Information data structure containing details about the Profound UI / Genie session indicated by the passed session id. See below for the DS definition. An escape message is sent if there are any errors.

## Job Information Data Structure

The job information data structure is defined as follows:

```
D JobInfo         DS              Qualified
D  Id                        16A
D  Name                      10A
D  User                      10A
D  Number                    10A
```