DesktopIcon Job Creation Process

For the job creation process in DesktopIcon, it is like this:

<1> Use **login()** API to obtain an accessToken.

<2> Use **getFileProfileInfoEx** API to get all File Profile information.

<3> Add file(s) into DesktopIcon and map to file profile. During mapping, available file profiles for one file are those which support this file’s extension. So if one source file has no extension, it can’t be mapped and be sent to server to create job from DesktopIcon at all.

<4> After all files are mapped and required options such as job name, target locales are entered, click “Submit” to begin creating job on GlobalSight server.

<5> DesktopIcon will upload files one by one using the **uploadFile(HashMap args)** API. It is an overwritten method for **uploadFile().**

If job attributes are setup, DesktopIcon will invoke **uploadAttributeFiles** API to upload job attributes files to server after all source files are uploaded.

<6> After all mapped files are uploaded to server, then use **createJob(HashMap args)** API to create a GlobalSight job. Job priority can be set with this API. Priority is a parameter in this method.

<7> If comment and/or reference file is specified, DesktopIcon will add them after createJob is completed on GS server.

DesktopIcon will invoke **getStatus()** API continuously to see if the job is created. Once job is created successfully, the “jobId” can be obtained from the returned string of this API. Then DesktopIcon will invoke **addComment()** API to add comment and/or reference file.

<8> To know which jobs are downloadable, DesktopIcon will invoke **getDownloadableJobs().** The jobs’ names client needs to know if they are downloadable are formed in xml and input as a parameter. In the returned xml, downloadable jobs are marked “downloadable” as status.

<9> After get the downloadable jobs’ names, DesktopIcon will invoke **getLocalizedDocuments()** API to get the localized files’ url. The job name as a parameter is okay. The returned xml contains the file url for downloading. Then common java can handle this.