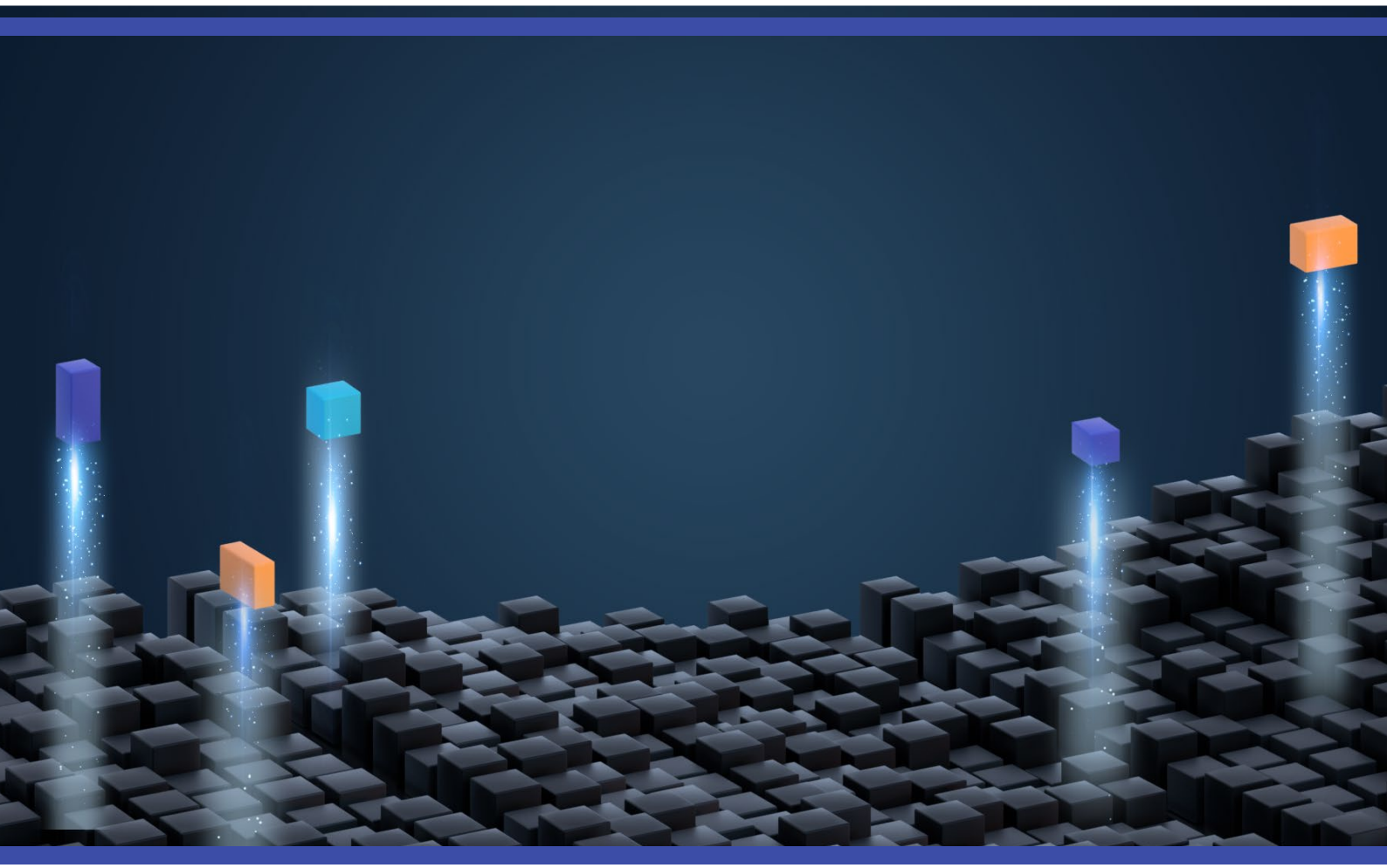
OpenLegacy
Digital to the Core

OpenLegacy Batch Job Submission October 2022



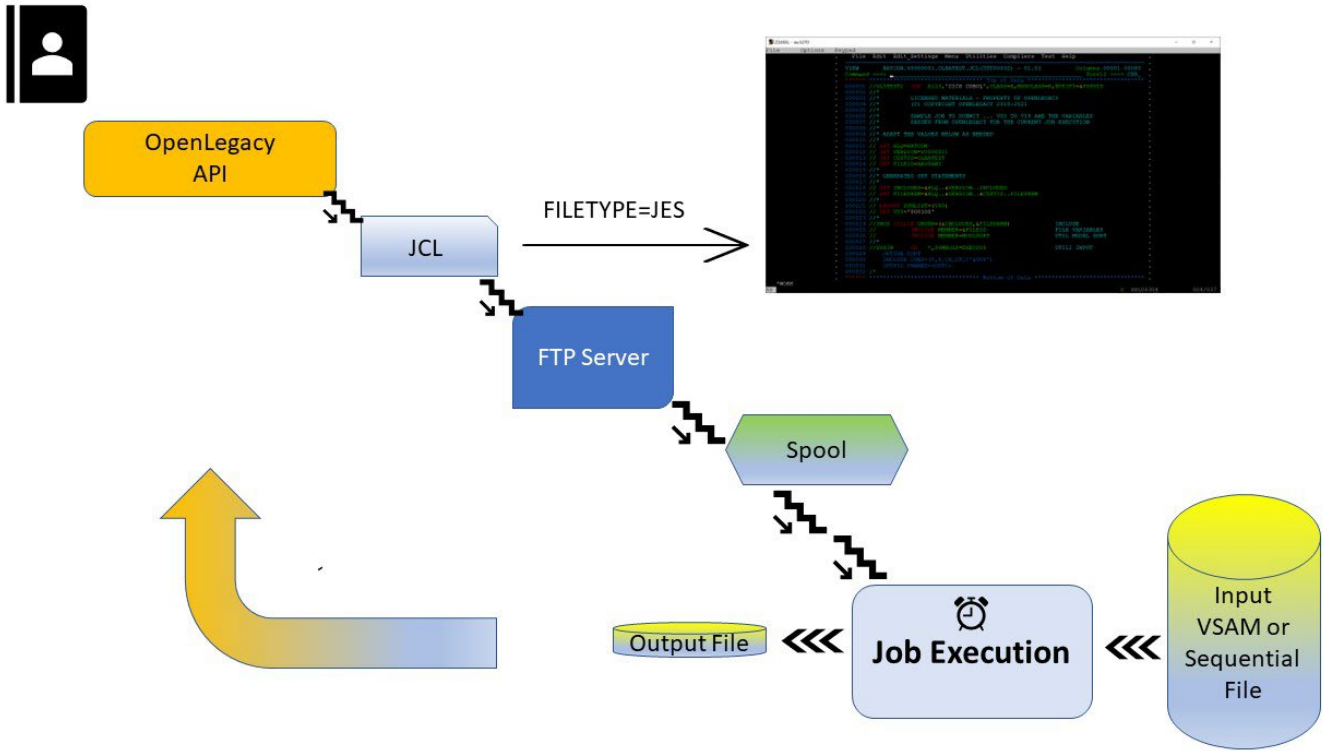
Contents

1 Introduction

This document describes the preparations required on the mainframe side to enable the submission of mainframe jobs from OpenLegacy APIs. The OpenLegacy APIs submit jobs when, for example, it is required to extract information from a sequential or VSAM file on the mainframe.

2 Flow

The diagram shows the submission of a job, its execution, and the creation of the output file with the response to the OpenLegacy API.



3 Summary of tasks required

For a successful project several actions are required by different teams. The table below describes them

Team	Action
z/OS System Programmer	Choose an HLQ for the libraries (an ALIAS may be required).
z/OS System Programmer	Copy the JCL/INCLUDE/FILEPARM libraries
z/OS System Programmer	Adapt INCLUDE JCL to site standards (if required)
Network Administrator	Create the Firewall rules
Security Administrator	Define RACF profiles for FTP and port protection
Security Administrator	Define RACF profiles for Datasets protection (both the input files to be read, and to the output files to be created)
Security Administrator	Create one/several UserIds for the execution of the jobs
Security Administrator	Create Keyrings and Server+Client Certificates for SSL protection
Security Administrator	Review/Approve end-to-end Flow
z/OS Communications Specialist	Customize FTP server for job submission
z/OS Communication Specialist	If required, define the port in the TCP PROFILE and request Firewall rules
z/OS Communication Specialist	If required, define the AT-TLS rules for SSL
z/OS Application Programmer	Create FILEPARM members with the metadate of the files to be processed
z/OS Application Programmer	Set up JOB card sample on the API

4 Logic

The submission of the JCL is based on the Mainframe FTP Server's ability to write a received file to the JES Internal reader. During the job execution, the job will create an output file which includes the JOBID as part of its dataset name, and such created file will be retrieved by the API at job termination. For example, if the purpose of the job is to extract records from a file, the output file will contain the selected records.
The API will open a passive connection to the FTP server. It is recommended to use SSL for protection, either via the FTP server itself, or via AT-TLS.

The API will poll the status of job execution to determine when the file is ready and will verify the $HASP395 message that indicates the job executed successfully, exists in the log.

To summarize, the APIs submit jobs exploiting facilities on the mainframe described at
https://www.ibm.com/docs/en/zos/2.3.0?topic=ftp-interfacing-jes

Behind the scenes, the API code will programmatically execute something resembling
https://www.ibm.com/docs/en/zos/2.3.0?topic=jes-examples
by submitting a command to the FTP server that contains a command like the one below
QUOTE SITE FILETYPE=JES JESJOBNAME=*
before executing the FTP PUT command to submit the JCL.

The API then pulls the file created by the job. The API uses the JES JOBID as a part of the DSNAME to identify the specific response file related to a specific API request.

5 How to Prepare the FTP Server

It is possible to either use an existing FTP server or to start a dedicated FTP server on a different port for the purpose of accepting JCL from an external source (the OpenLegacy API in this case).

Using a different FTP server with a different port can simplify Firewall and Security definitions.

The sample parameters shown below detail how to enable the FTP server to accept the SITE FILETYPE=JES FTP commands that will be issued by the OpenLegacy API SDK to the FTP server.

```
JESLRECL       80              ; lrecl of jes jobs
JESPUTGETTO    600             ; timeout for remote job submission put/ge
JESRECFM       F               ; recfm of jes jobs
JESINTERFACELEVEL 2            ; Allow arbitrary jobnames
```

If implementing SSL via AT-TLS, pay attention to parameters SECURE_LOGIN and VERIFY_USER to properly protect your system according to your policies.

6 Sample JCL for Dataset Processing

A common use of the batch facility is to extract data from mainframe files (such as VSAM, seq, etc.).
Sample JCL used in the OpenLegacy Lab is provided as a basis for creating your local JCL.
The JCL is organized in libraries that allow its re-use and its compactness.

Three PDS libraries should be created, whose LLQs are
INCLUDES
FILEPARM
JCL

The INCLUDES library contains fragments of JCL that, when combined, will compose the JCL of the job actually submitted.

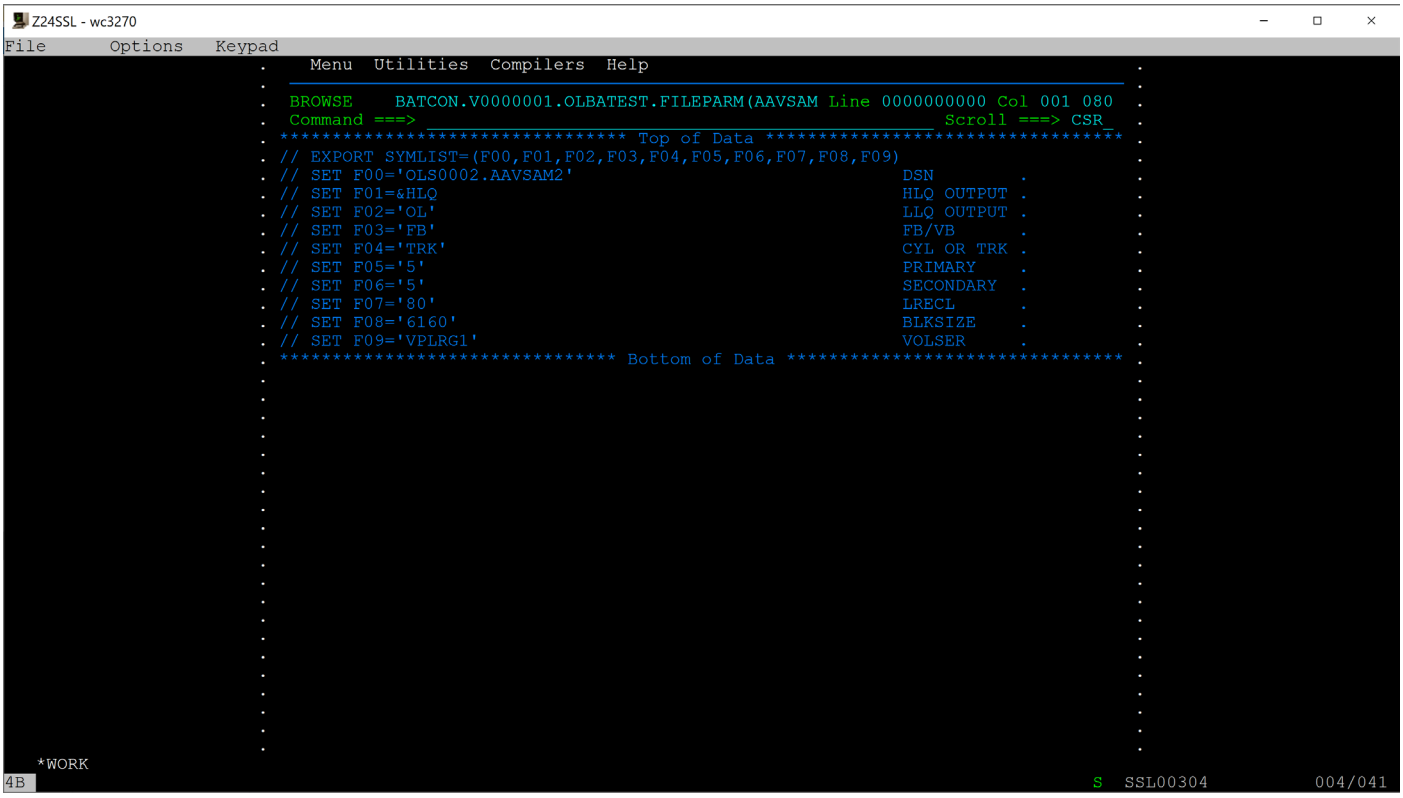
The FILEPARM library contains the parameters to process a specific file, such as LRECL, BLKSIZE, etc.

The combination of the FILEPARM and INCLUDES creates the JCL for a job.

The JCL library contains sample jobs similar to those that will be triggered by the API side (those that the API will put on the JES INTRDR).

7 Preparing FILEPARM Members

The z/OS application programmer needs to create a FILEPARM member for each input file to the job.
In the sample below you can see that variable F00 represents dataset names, while others help to specify the name of the created output file or its DCB characteristics.



8 Summary for Success

The firewall needs to allow the FTP requests to flow from the API to the mainframe (this involves the network team).

The distributed snippet JCL and FILEPARM descriptions need to be copied and adapted on the mainframe libraries.

The Security team will need to give the necessary RACF commands to:

Permit the submission of JCL from the API and the reading of the response files by the API.

Permit the reading of the input VSAM and/or sequential files, and the writing of the output files by the submitted jobs.

It may be necessary to create a new Mainframe USERID under which the submitted jobs will run, and probably SSL certificates for the purpose of Client Authentication. For a POC, an existing USERID and Basic authentication can be used (USERID and Password)

Permit access to all the Security Profiles to use the FTP services the API requests. See “Security Profiles to Look At”

The mainframe team will need to adapt the provided snippet JCLs to the site standards (naming conventions, HLQs, VOLSER, SORTWORK space, etc.)

The mainframe team will need to create the members in the FILEPARM library used to specify the DSNAME and DCB characteristics of the files we are processing (e.g., DSNAME, LRECL, BLKSIZE, etc.)

9 Security Profiles to Look At

Customers protect their systems using different RACF profiles (or their equivalent Top Secret or ACF2 definitions). OpenLegacy cannot provide a complete list of RACF definitions to create because such list depends on the actual customer setup for profiles and the profile classes activated.
Following you can find some profiles that are candidates to produce ICH408I messages.
Refer for the IBM documentation for the FTP Server Setup planning and reference.

Class SERVAUTH
EZB.FTP.*.*.ACCESS.HFS (G)
EZB.FTP.*.*.PORT2021 (G) e.g. EZB.FTP.sysname.ftpdaemonname.PORTnnnnn

Class FACILITY (for a new FTP server)
For USS
BPX.DAEMON
BPX.SERVER
BPX.SUPERUSER

For certificates
IRR.DIGTCERT.*

10 FTP Server Parameters to Look At

OpenLegacy cannot provide a complete list of parameters which depend on the actual customer setup,
Following are some FTP parameters that are candidates to be relevant.
Refer for the IBM documentation for the FTP Server Setup planning and reference.

For AT-TLS

```
TLSMECHANISM   ATTLS
TLSPORT
```

```
SECURE_LOGIN
VERIFY_USER
```

Non AT-TLS

```
TLSMECHANISM    FTP
SECURE_FTP      REQUIRED
SECURE_CTRLCONN PRIVATE
SECURE_DATACONN PRIVATE
CIPHERSUITE     SSL_AES_256_SHA
KEYRING         FTPSSL.Server
EXTENSIONS      AUTH_TLS
DEBUG           SEC
```

Other Parameters Worth Review

```
SECURE_LOGIN
VERIFY_USER
Everything starting with SECURE_

SMFJES
FWFRIENDLY  TRUE              ; client to work across NAT and VPN
PASSIVEDATACONN
PASSIVEDATAPORTS
JESENTRYLIMIT 200
```