

Incorporating Pinnacle21[®] With LSAF[®]

Sonali Garg, Alexion Pharmaceuticals, New Haven, USA

Sandeep Juneja, SAS Institute, Cary, USA

Aleksey Vinokurov, Pinnacle21, Plymouth Meeting, USA

ABSTRACT

Clinical studies data need to adhere to CDISC standards for submission to Regulatory agencies and to check if data is compliant to these standards, Pinnacle21 (P21) checks seem to be the de facto standard widely used by companies.

Life Science Analytics Framework (LSAF) provides ability to have 21 CFR part 11 compliant web based Statistical Computing Environment (SCE) where studies data can be stored securely with versioning and audit history capabilities. However, since data had to be downloaded and moved manually to P21 site to execute compliance checks, traceability of the data was at risk.

This paper talks about steps taken to automate integration between LSAF and P21, thereby eliminating manual data movement process. Using P21 client, P21 validator can be invoked from LSAF directly which handles the data movement and execution of the checks and results report of the checks are checked into LSAF Environment thereby maintaining full traceability and integrity of the data.

LIFE SCIENCE ANALYTICS FRAMEWORK (LSAF)

Life Science Analytics Framework (LSAF) is a web-based Statistical Computing Environment (SCE) that provides functionalities like version control, permission access and audit history. It allows users to execute SAS programs using concept of Jobs that provide ability for impact analysis.

P21 CLIENT APPLICATION

Users of the P21 Client application can create projects, studies, and data packages on the P21 Enterprise Server and run validations from command line. P21 Server can be integrated with LSAF which allows seamless data transfer from LSAF repository onto P21 Server. Users can provide data package parameters right from the command line to configure validation to any specification that is available on P21 Server. After launching validation process P21 Client waits for completion of the validation job and downloads Excel validation report. P21 Client application evaluates results of validation and returns severity code (1 – Rejects, 2 – Errors, 3 – Warnings, and 4 - Notice) back to command line.

P21 CLIENT LSAF INTEGRATION SCENARIO

P21 Client application is a Java jar and due to security reasons LSAF does not allow command executions from Command prompt. In order to support execution of Java jar from SAS program, SAS provides JavaObj functionality.

Any java jar classes can be invoked from SAS program using JavaObj, supporting execution of Java code from SAS program. This allows full integration of the P21 Client features into LSAF environment.

SAS program/ SAS Job can be written within LSAF to launch a data set validation without any intermediate data transfers between servers and systems. Validation process can be fully automated inside LSAF ecosystem and can be standardized with a set of SAS macros and programs.

THE BRIDGE

P21 Enterprise Server has a plugin architecture that allows dynamic configuration of various data adapters. To enable LSAF integration Pinnacle 21 developed LSAF adapter that is based on **lsaf-java-api-client**. LSAF Java API provides easy access to all Repository services thus enabling data transfer from LSAF Server onto P21 Server. To minimize data transfer time and network traffic P21 Adapter utilizes zip services of the LSAF Server compressing data before transferring it onto P21 Server.

SAS PROGRAMMING

javaobj API makes it easy to run P21 Client application from the SAS program:

```
dcl javaobj jCmd("net/pinnacle21/oce/utils/ProcessUtils");
length exitCode 8.;
array s{&i.} $200 (&cmd.);
jCmd.callStaticVoidMethod("startFromCommandLine",s);

jCmd.callStaticIntMethod("exit", exitCode);
put exitCode=;
```

Here (&cmd.) parameters that SAS program provides to configure P21 validation process. The following is an example of calling SAS procedure with parameters for P21 Client.

```
%let study_path = %str([study path]);
%let report_path = %str(&_SASWS_/[output path]);
%let report_name = Result_%SYSFUNC(translate(%SYSFUNC(datetime()),
E8601DT.),'-',':');
%let version = %str(pinnacle21-cli-1.0.0);
%put &report_name;
```

* Validate datasets;

```
%pinnacle21(version=&version..jar,
  Params=%nrquote(
    --project=Project name,
    --study=Study,
    --standard=sdtm,
    --standard.version=3.2,
    --meddra.version=19.0,
    --cdisc.ct.sdtm=2017-03-31,
    --source.sdtm="lsaf:///&study_path",
    --report=&report_path./&report_name..xlsx,
    --report.cutoff=10),
  report_path=&report_path.,
  debug=N);
```

These parameters are enough to launch dataset validation on the P21 Enterprise Server. Latest versions of other dictionaries that are loaded on the P21 Server will be used if not provided on the command line. For complete reference of all available parameters see **P21 CLIENT REFERENCES**. Please note **lsaf:///** prefix for **--source.sdtm** parameter. This is data source schema that enables data transfer from LSAF Repository onto P21 Server. Report parameter **--report** specifies local LSAF Repository Excel report file that will be created by the P21 Server validation process and transferred onto LSAF Server by the P21 Client application. **--report.cutoff** parameters helps control the size of the validation report. Please note that P21 Enterprise Server will contain the full (without **--report.cutoff** limit) dataset validation result details and users will have access to more in depth validation analysis.

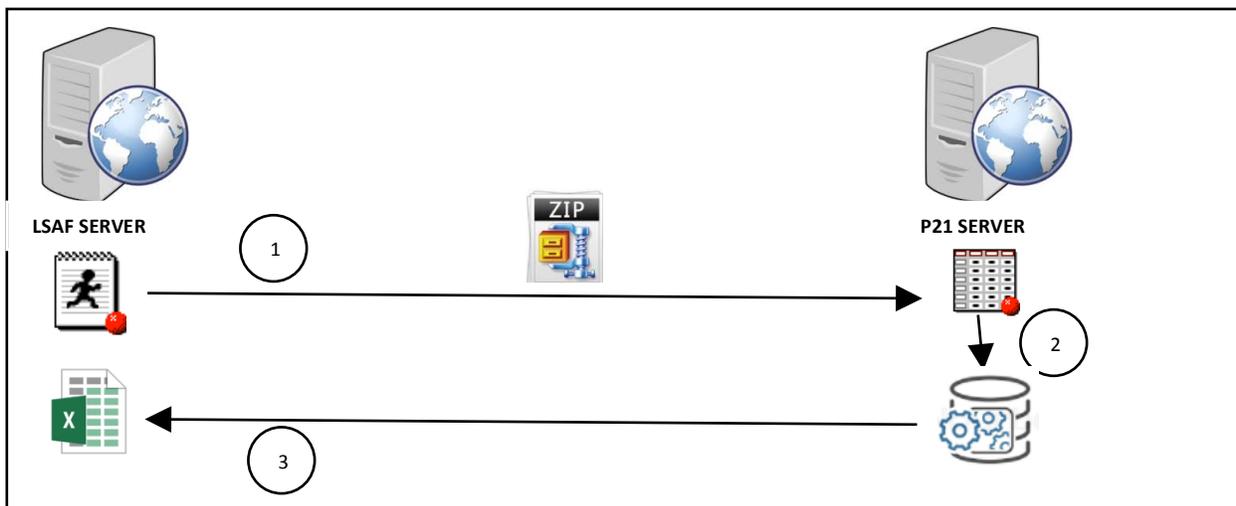
P21 CLIENT FILES

There are two files that need to be loaded into LSAF repository; *pinnacle21-cli-1.0.0.jar* and *pinnacle21.conf*. Although all parameters can be passed in on the command line it is recommended to place static configuration that is not expected to change from one study to the next into a *pinnacle21.conf* file to make client call more concise.

Example of the *pinnacle21.conf* file:

```
web.host="[https://[p21 server].pinnacle21.net]"
api.key=API access key provided by the P21 Enterprise Server Admin. P21 Enterprise Server will be configured with an API key]
group=[user group. Typically, P21 Server will be configured with default user group called enterprise_cli]
# These parameters are default configuration for HTTP connections
default.http.max.connections = 10
default.http.max.connectTimeout = 5000
default.http.max.requestTimeout = 5000
default.http.max.socketTimeout = 10000
#datapackage client
datapackage.http.max.socketTimeout = 15000
# validator client
validator.http.max.socketTimeout = 15000
# define client
define.http.max.requestTimeout = 10000
define.http.max.socketTimeout = 15000
# To enable HTTPS connectivity between LSAF and P21 Server
# the following configuration is needed:
default.http.proxy.host="[proxy server URL]"
default.http.proxy.port=[port number]
```

LSAF PINNACLE 21 CHECKS EXECUTION FLOW:



Above diagram summarizes the flow involved in execution of the checks

1. SAS Program/ SAS Job is executed within LSAF controlled Environment which authenticates and connect to Pinnacle 21 Server. Data is securely transferred in compressed format to P21 Server.

2. Data is extracted on P21 server and checks are executed.
3. Final Excel report generated via execution checks is transferred back to LSAF environment.

ACCESS CONTROL AND SECURITY

P21 Enterprise users continue to enjoy access control and security features of the Enterprise Server. After studies are automatically created from the SAS program, project managers can assign access rights to users of the P21 Enterprise Software.

Integration

1. P21 Server administrator creates an API key and special user group called **enterprise_cli**. P21 Client application will use this key to automatically create or update projects and studies and to run validations. It is recommended to store this key in the *pinnacle21.conf* file mentioned above as **api.key**. API user group is also stored in the same conf file as **group = enterprise_cli**.
2. LSAF administrator creates a user with Read access to all study data. P21-LSAF adapter securely stores LSAF user credentials on the P21 Server.

CALLING PINNACLE 21 FROM WITHIN LSAF

We created a global program which is saved in a centralized location accessible to all LSAF users. Study specific macros will call this global program from the study specific location and the validation report is also saved to the study area. This way it respects all the permissions and only people who are part of the study can see the calling program and the validation report.

Once the SAS Program/SAS Job is called from LSAF, the data is transferred from LSAF to P21, validation is run on it, excel report is produced and pushed back onto LSAF.

If the user wants to access the same report on P21 url, they should have appropriate credentials for it. Below is the header portion of the report produced on LSAF listing all the versions of the file used in validation.

Pinnacle 21 Enterprise Validation Report	
Generated:	2017-11-02T13:49:09
Standard:	SDTM-IG 3.2
CDISC CT Version:	2017-03-31
MedDRA Version:	20.1
SNOMED Version:	2016-09-01
UNII Version:	2016-09-06
NDF-RT Version:	2016-10-03
Software Version:	3.2.2

CONCLUSION

The above process provides a mechanism to manage and maintain data securely within LSAF and providing a mechanism to execute Pinnacle 21 Checks in a secured way by securely integrating two independent systems

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Sonali Garg
Alexion Pharmaceuticals
100 College St,
New Haven, CT 06510
Phone: +1-919-641-8678
Email: Sonali.Garg@alexion.com

Sandeep Juneja
SAS Institute Inc
801 SAS Campus Drive
Cary, NC 27513
Phone: +1-919-531-0541
Email: Sandeep.Juneja@sas.com

Aleksey Vinokurov
Pinnacle 21 Inc
31 Plymouth Rd #508,
Plymouth Meeting, PA 19462
Phone: +1 (888) 507-2270
Email: avinokurov@pinnacle21.net

APPENDIX:

P21 CLIENT REFERENCES

Parameter	Valid Values	Req	Default	Description
Validator - Source Data				
--source.adam	<path>			Path to source ADaM data files. Can contain directories or individual files separated by semicolon.
--source.sdtm	<path>			Path to source SDTM data files. Can contain directories or individual files separated by semicolon.
--source.send	<path>			Path to source SEND data files. Can contain directories or individual files separated by semicolon.
Standard and Config				
--standard	SDTMIG SDTMIG-MD SDTMIG-AP SDTMIG-PGx SENDIG ADaMIG	x		Standard to validate against.
--standard.version	<version>	x		Version of standard.
--filter	PMDA FDA ALL		ALL	Config filter to apply.

Validator - Terminology				
--cdisc.ct.adam.version	<version>		<latest>	Version of ADaM CT controlled terminology.
--cdisc.ct.sdtm.version	<version>		<latest>	Version of SDTM CDISC controlled terminology.
--cdisc.ct.send.version	<version>		<latest>	Version of SEND CDISC controlled terminology.
--meddra.version	<version>		<latest>	Version of MedDRA dictionary.
--ndf-rt.version	<version>		<latest>	Version of NDF-RT dictionary.
--snomed.version	<version>		<latest>	Version of SNOMED dictionary.
--unii.version	<version>		<latest>	Version of UNII dictionary.
Validator - Report				
--report	<file>			Path and file name to where the report will be saved.
--report.cutoff	<#>		1000	Number of times a record detail is printed per issue. For no limit specify 0.