

ISPP, an Integrated Statistical Programming Platform to systematically manage study programming environment, program, output, log, lst and more

Jinling.Li, BeiGene

ABSTRACT

Manage study programming environment and relevant files could be very complicated and time-consuming:

Analysis directories usually have specific setting or initialization associated with difference level of access control. Switching from one analysis directory to another is not only changing the folder, but also setting up a new environment.

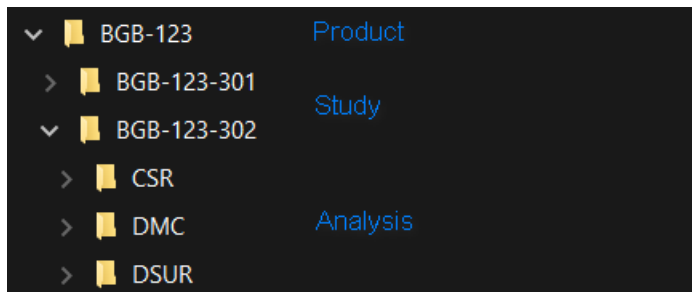
An analysis directory may have hundreds of files including programs, outputs, log, lst and many other documents. Batch running a bunch of programs, verifying log / lst file, and making sure outputs, datasets and relevant files are in a chronological order as expected are programmers' daily work.

ISPP is a JavaScript based tool which aims to manage those challenges in a systemic way: ISPP makes switching between analysis folder a lot easier and provides programmers a user- friendly interface to batch run program etc. Furthermore, ISPP provides a dashboard to monitor batch run status and show alerts if any issues in log / lst, or relevant files are not in planned chronological order.

ISPP also have some utilities that can improve programming efficiency such as scheduling batch run, merging PDF, creating program verification form, integrating with program tracker etc. All those features will be described in this paper.

INTRODUCTION

In clinical programming industry, to better manage SAS programming activities, we usually organize the programs, deliverables by triplet of product, study, and analysis:



Each triplets have some specific setting such as global macro variable for address, SAS library, SAS options, etc., programmers usually work on multiple triplets and need to switch triplets quite often, how can we easily switch triplet with corresponding setting correctly applied? Is there a way to show programmer the current working address once we get SAS enterprise guide or SAS system opened?

Batch running programs, checking log, verifying lst and making sure outputs, datasets and relevant files are in a chronological order are critical part of quality control of programming deliverables, number of SAS conference papers, presented several innovative solutions by VBA, SAS/AF or Python, each solution usually focus on specific functionality but not all, can we have an integrated tool for all functionalities we need?

In this paper we would like to present an integrated solution with visualization and interactive functionalities by JavaScript, by this tool, we can address all questions above.

INITIALIZE ALL SETTINGS FOR WORKING FOLDER

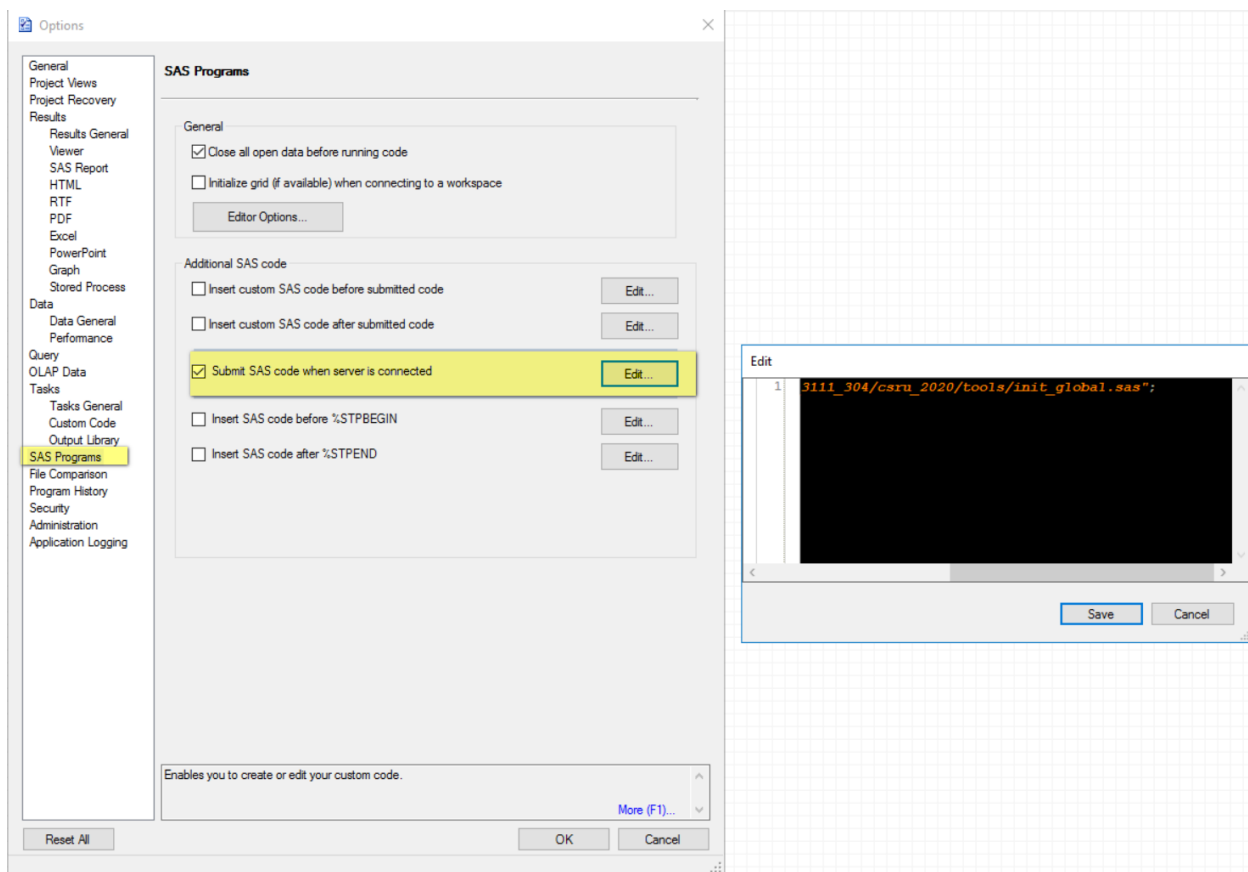
Within each triplet, there are two initialization files: `init_global.sas` and `init_local.sas`, those two files are created up on creation of triplet.

`init_global.sas` is read-only, it initializes standard global macro variable of path, SAS library, SAS options, file reference for triplet, it also includes `init_local.sas` in the end.

`init_local.sas` can be edited by user but cannot be deleted by user, it contains some specific definition of global macro variable of path, SAS library, SAS options, file reference, format for triplet.

RUN CODE IN INTERACTIVE MODE

To run program in a triplet and with all settings applied, we need to run `init_global.sas` first, but every time we run program, we must open `init_global.sas` and run it first, it is annoying right? If you are using SEG (SAS enterprise guide), then you can specify full address and name of `init_global.sas` in option of 'Submit SAS code when server is connected', which means, whenever you run program via SEG, `init_global.sas` will be executed just before running your submitted program. It looks better right? But what if we need to change our working triplet very often? Yes we need to update that option in SEG whenever we change working triplet, It is still a burden, but do not worry, ISPP can do this for you, you will find it in follow sections.



RUN CODE IN BATCH MODE

To run program by batch with all settings applied, we need to ask SAS server to firstly run `init_global.sas` in corresponding triplet, this can be achieved by SAS batch command option 'autoexec',

```
<path>/sas -autoexec /<path>/init_global.sas /<path>/my_program.sas
```

You can also specify where to output the log and lst by SAS batch command option 'log' and 'print'.

Yet again, whenever you need to batch run program you need to write the command line with that option, and submit the command line in terminal, or you can write some scripts for it, but you need to update the scripts whenever there are some new/deleted program or changes in program name, ISPP has solution for it.

LOG, LST, OUTPUT CHECKING

After batch run, we have to check log, verify lst and make sure outputs, datasets and relevant files are in a chronological order, I believe this can be done by writing a SAS macro, but ISPP can provide a real-time dashboard for it as soon as batch run complete.

WHAT ISPP CAN DO

As an integrated platform for SAS programmer, ISPP can:

- Triplet management
- SEG customization
- Batch run (with or without program tracker)
- Log, lst, output checking
- Triplet dashboard
- Other Utilities

TRIPLET MANAGEMENT

You can add the triplets you frequently work on in ISPP home page, for each triplet, you can specify the location of init_global.sas, the encoding (UTF8 or Latin) and other triplet level setting.

The screenshot shows the 'Analysis Management' interface in the ISPP. It features a search bar, filter options for Compound, Study, and Analysis, and a table of triplets. The table has columns for Select, Compound, Study, Analysis, and Actions. The Actions column contains buttons for Setting, Open SEG, Dir Mode, Tri Mode, and Dashboard. Below the table are 'Select all' and 'Clear selected' buttons.

Select	Compound	Study	Analysis	Actions
	bgb_3111	docs	adam	Setting Open SEG Dir Mode Tri Mode Dashboard
	bgb_3111	bgb_3111_304	ash_abs_202107	Setting Open SEG Dir Mode Tri Mode Dashboard
	bgb_3111	bgb_3111_304	csrb	Setting Open SEG Dir Mode
	bgb_3111	bgb_3111_304	csru	Setting Open SEG Dir Mode Tri Mode Dashboard
	bgb_3111	bgb_3111_304	csru_2020	Setting Open SEG Dir Mode Tri Mode Dashboard
	bgb_3111	bgb_3111_304	dmc_2021_ja	Setting Open SEG Dir Mode Tri Mode Dashboard
	bgb_3111	bgb_3111_304	dsur_ib_2021	Setting Open SEG Dir Mode
	bgb_3111	bgb_3111_304	idmc2_2021	Setting Open SEG Dir Mode Tri Mode Dashboard
	zzz_tools	m_arc	v_1_0	Setting Open SEG Dir Mode

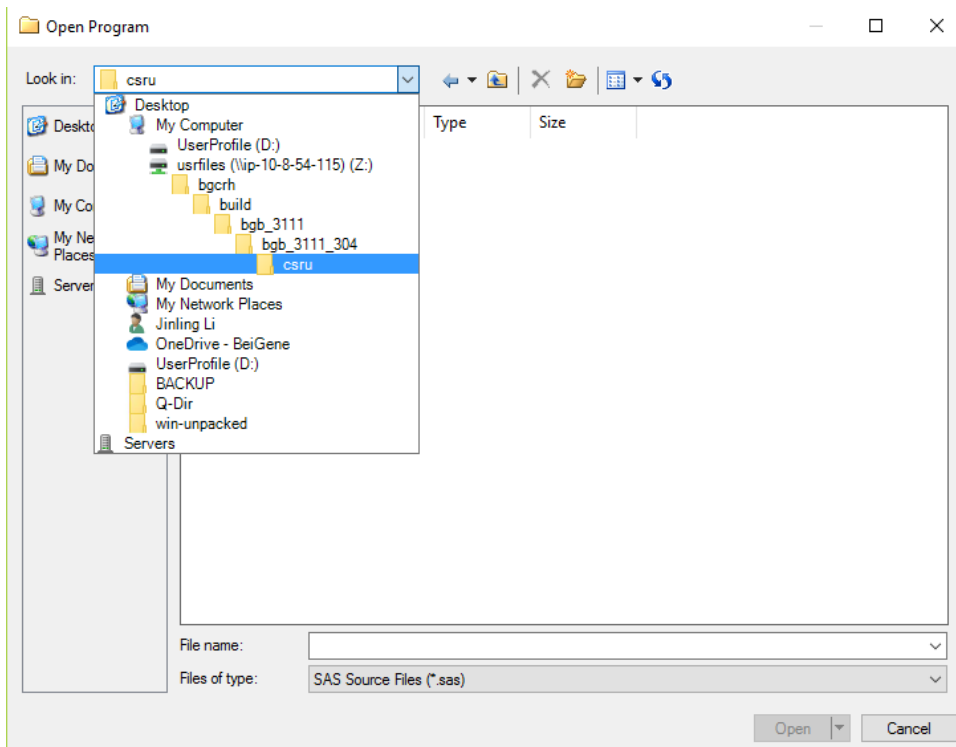
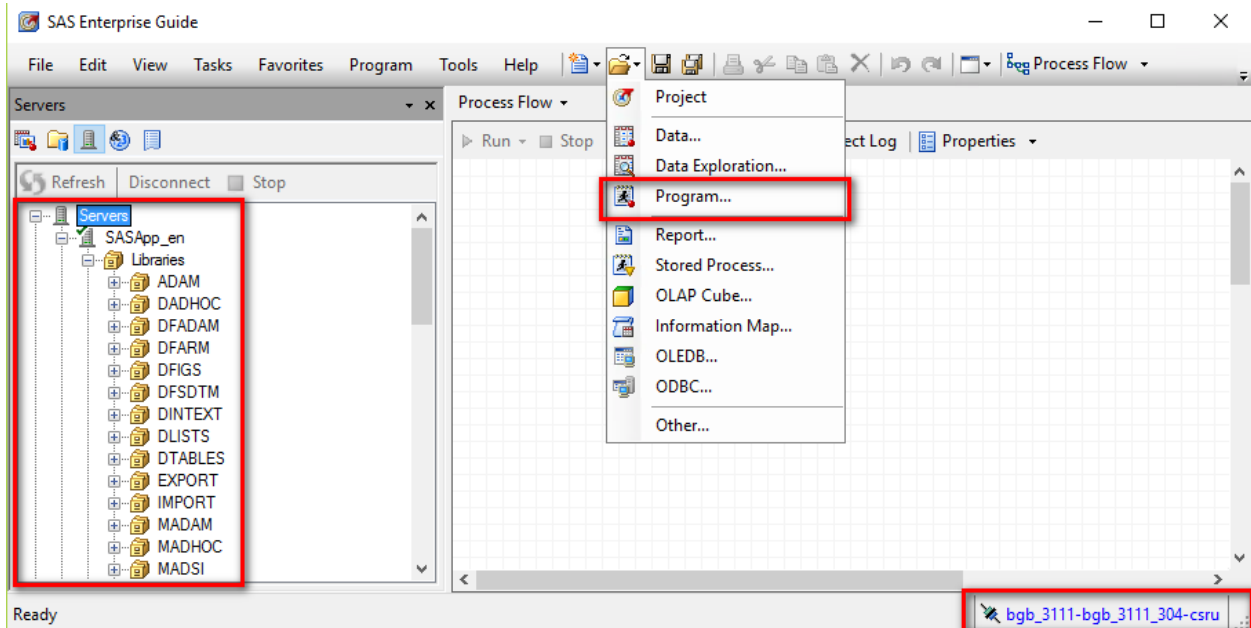
SEG CUSTOMIZATION

Once click 'Open SEG', ISPP will open SEG and apply all setting of triplet:

ISPP configures the option of 'Submit SAS code when server is connected' in SEG for corresponding triplet on which you click 'Open SEG'.

ISPP configures SEG to show triplet name which are you currently working on.

ISPP also sets the default folder for 'Open Program' dialog window as your current triplet folder, so you do not need to search the program from root folder.



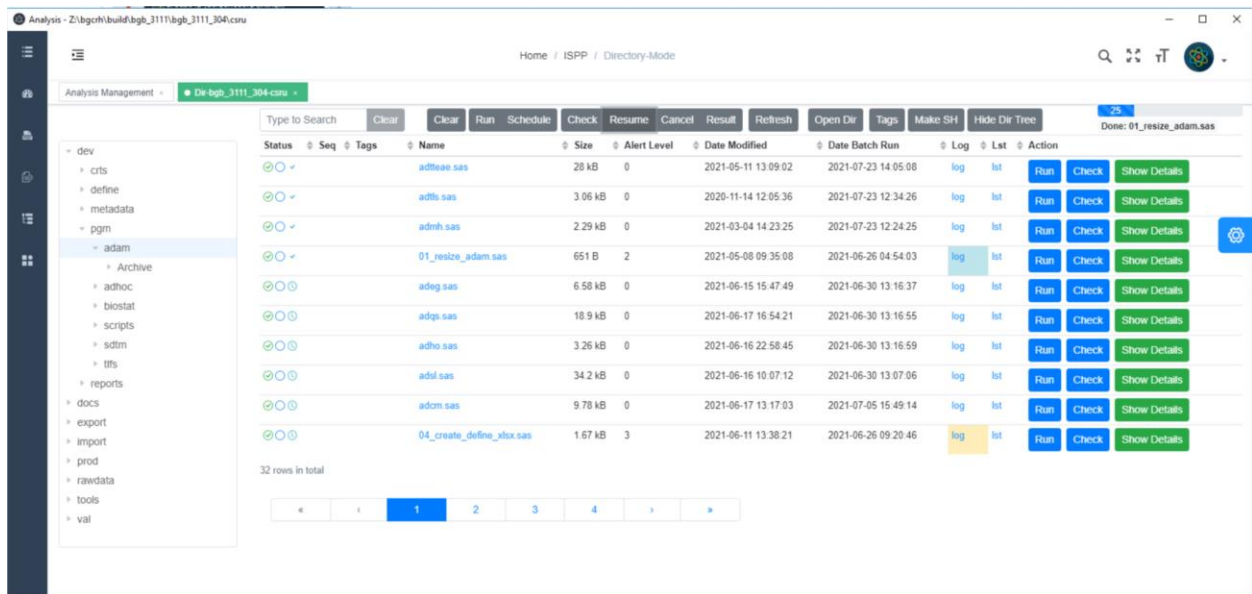
BATCH RUN

ISPP provides an easy to batch run program, and provides real-time progress and status, you can browser files in the analysis folder via ISPP, select multiple programs, customize the batch run sequence, you can even suspend and resume the batch run queue, and of course you can schedule the batch run job.

ISPP checks the log and lst immediately once the batch run completed for each program, issue in log or lst is indicated by different color, issue details can be found in 'Show Details'.

By clicking the program name, log, lst, you can open the corresponding file, this gives user the convenience, rather than searching the file in windows explorer.

ISPP does not check log/lst if batch run program by schedule mode, but user can check the program's log lst without batch run.



ISPP can also work with program tracker, after program tracker uploaded, you can easily to batch run program, check log/lst, for selected entries of ADaM, SDTM or TLFs.

All columns in tracker can be added in ISPP interface, user can use those columns for filtering or searching.

Again, any issues in log, lst, timestamp is indicated by different color, for example, lst in red means compare does not match or no compare information found; log in red means error in log, while log in yellow means warning and blue means special note.

Analysis - bgh_3111-bgh_3111_304-ciru

Home / ISPP / Tracker-Mode

Analysis Management - tik bgh_3111_304-ciru

ADAM

Column Filter Search Clear Refresh Tool Run Schedule Check Detail

Status	SEQ	PGM	VPGM	Alert	Dtime	Vtime	Log	Vlog	Vst	Action
✓	0	adsl.sas	v_adsl.sas	4	2021-07-23 12:22:57	2021-07-23 12:41:09	log	vlog	vst	Run Check Detail
✓	1	adrs.sas	v_adrs.sas	3	2021-07-23 12:23:03	2021-07-23 12:41:39	log	vlog	vst	Run Check Detail
✓	2	adae.sas	v_adae.sas	0	2021-07-23 12:24:09	2021-07-23 12:41:49	log	vlog	vst	Run Check Detail
✓	3	adcm.sas	v_adcm.sas	0	2021-07-23 12:24:23	2021-07-23 12:41:58	log	vlog	vst	Run Check Detail
✓	4	adnh.sas	v_adnh.sas	0	2021-07-23 12:24:25	2021-07-23 12:42:00	log	vlog	vst	Run Check Detail
✓	5	adteea.sas	v_adteea.sas	4	2021-07-23 12:24:33	2021-07-23 12:42:38	log	vlog	vst	Run Check Detail
✓	6	adfb.sas	v_adfb.sas	2	2021-07-23 12:31:51	2021-07-23 12:51:01	log	vlog	vst	Run Check Detail
✓	7	adlby.sas	v_adlby.sas	0	2021-07-23 12:32:08	2021-07-23 12:51:27	log	vlog	vst	Run Check Detail
✓	8	adex.sas	v_adex.sas	0	2021-07-23 12:32:17	2021-07-23 12:51:33	log	vlog	vst	Run Check Detail
✓	9	adexsum.sas	v_adexsum.sas	4	2021-07-23 12:32:30	2021-07-23 12:52:21	log	vlog	vst	Run Check Detail
✓	10	advs.sas	v_advs.sas	0	2021-07-23 12:32:41	2021-07-23 12:52:33	log	vlog	vst	Run Check Detail
✓	11	adecog.sas	v_adecog.sas	0	2021-07-23 12:32:49	2021-07-23 12:52:39	log	vlog	vst	Run Check Detail
✓	12	adeg.sas	v_adeg.sas	0	2021-07-23 12:32:51	2021-07-23 12:52:43	log	vlog	vst	Run Check Detail
✓	13	adlr.sas	v_adlr.sas	3	2021-07-23 12:33:00	2021-07-23 12:52:55	log	vlog	vst	Run Check Detail
✓	14	adlte.sas	v_adlte.sas	3	2021-07-23 12:33:02	2021-07-23 12:52:58	log	vlog	vst	Run Check Detail

26 rows in total

First Prev 1 2 Next Last

ARCHITECTURE OF ISPP

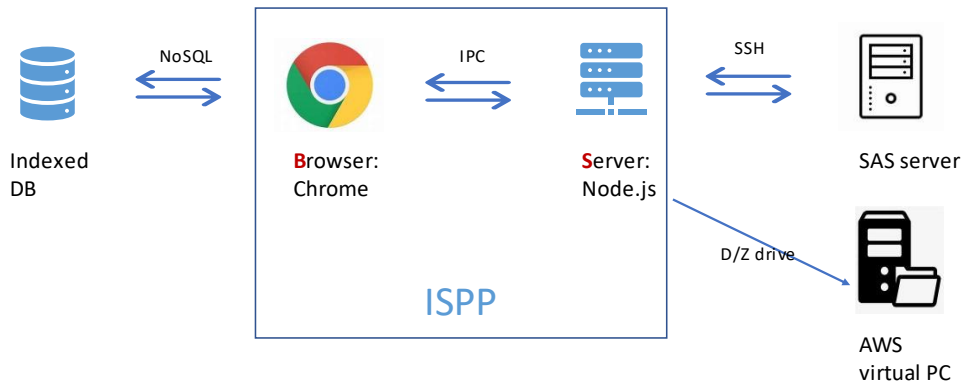
ISPP is developed by JavaScript, with below architecture, Indexed DB, a NoSQL database, provides the data storage service in frontend.

Within framework of Electron, there are two components, Chromium and Node.js, Chromium is the interface for ISPP, while Node.js is the server end, they can communicate via inter process communication (IPC).

User's data is stored in Indexed DB which can communicate with Chromium, Node.js can communicate with SAS server via either SSH or HTTPS. Node.js can also communicate with user's local PC, no matter it is an AWS virtual or real laptop.

The general workflow of ISPP is, retrieve program information from user's local PC, submit program information to SAS server, perform checks and store information in Indexed DB, present the information to user by Chromium.

ISPP, an instance of Electron



CONCLUSION

ISPP provides an integrated platform for programmers' daily work, but still some limitation: all data is stored in user local PC, no centralized monitor of users' batch job and cannot performs the batch job balancing.

Also, as PC application, user need to upgrade it manually if any new version of ISPP released.

ACKNOWLEDGMENTS

I would like to offer a special thanks to You Li and Quting Zhang, providing a technical review of the issues brought up in this paper.

CONTACT INFORMATION <HEADING 1>

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

Jinling.Li
BeiGene, Ltd.
Jinling.Li@beigene.com