Current Workflow to add a new test plan to Online Regression

Friday, September 04, 2015 8:00 PM

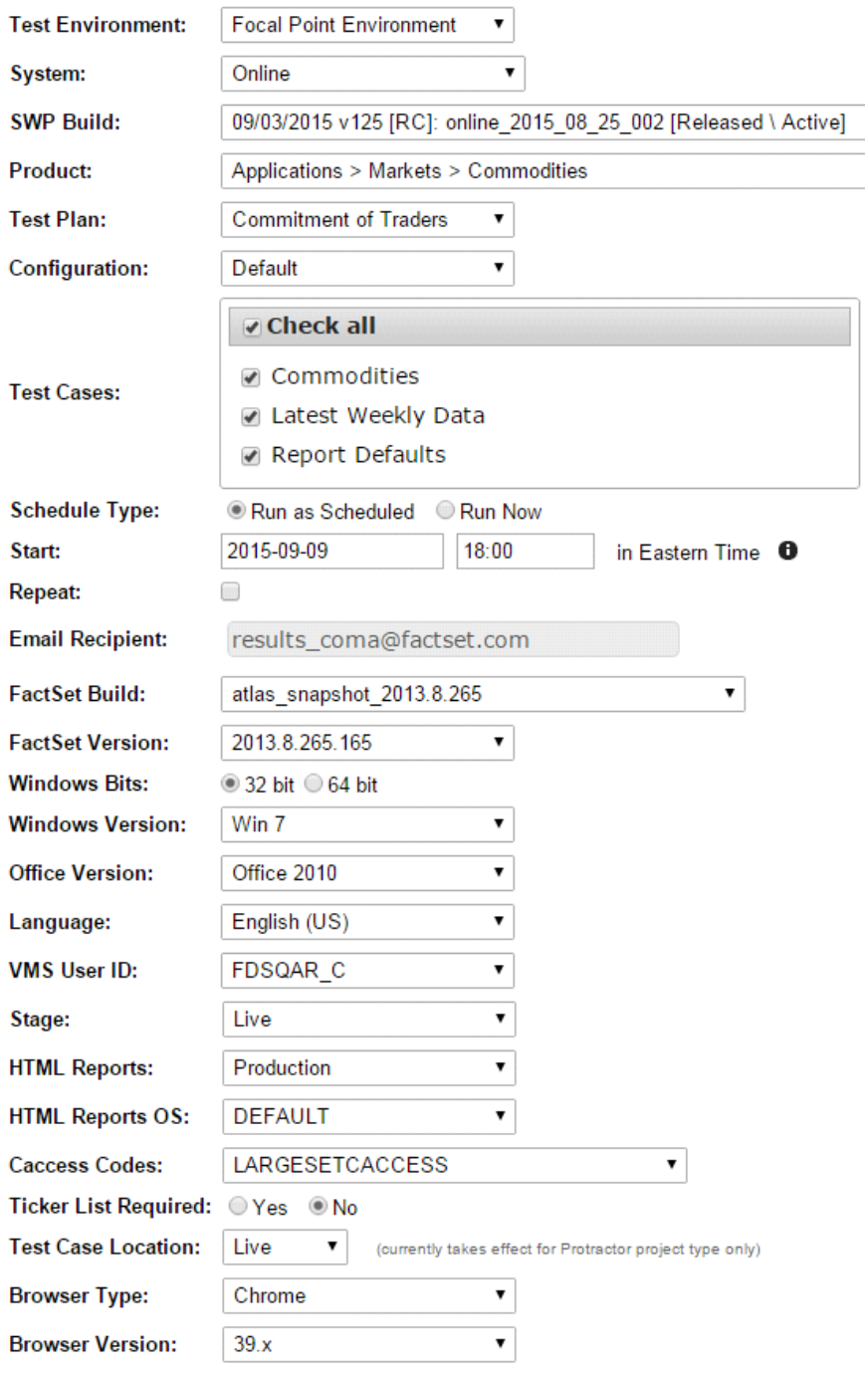
It starts with an RPD from a QA lead of a particular test plan. An example rpd - http://is.factset.com/rpd/summary.aspx?messageid=19004370

For each request,

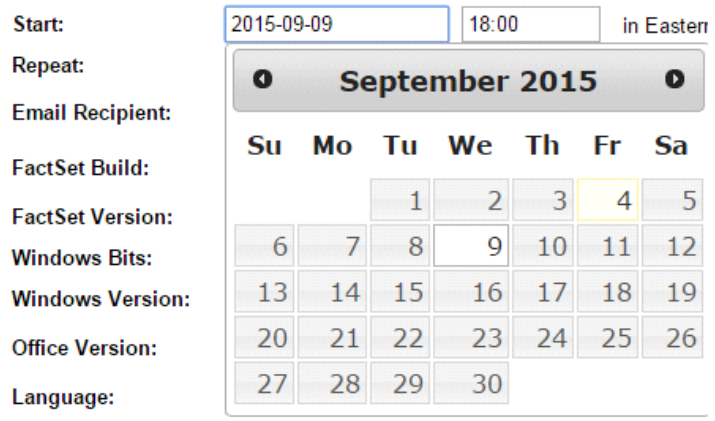
- We make sure that all the required info to schedule the job is provided in the RPD. If those are not provided, we request them to provide the same.
- Once we have all the info, we make an entry for the test plan in the Online Cron Schedule spreadsheet. An example entry looks like:

Commitment of Traders	Online- 2013.9	FDSQAR_C	Devel	Devel	LargesetCaccess	18:00 ET Wed	Default
Commitment of Traders	Online- 2013.9	FDSQAR_C	Devel	Staging	LargesetCaccess	18:20 ET Wed	Backwards Compatibility

- Then we go to the JEP and start filling in the form with the provided details:



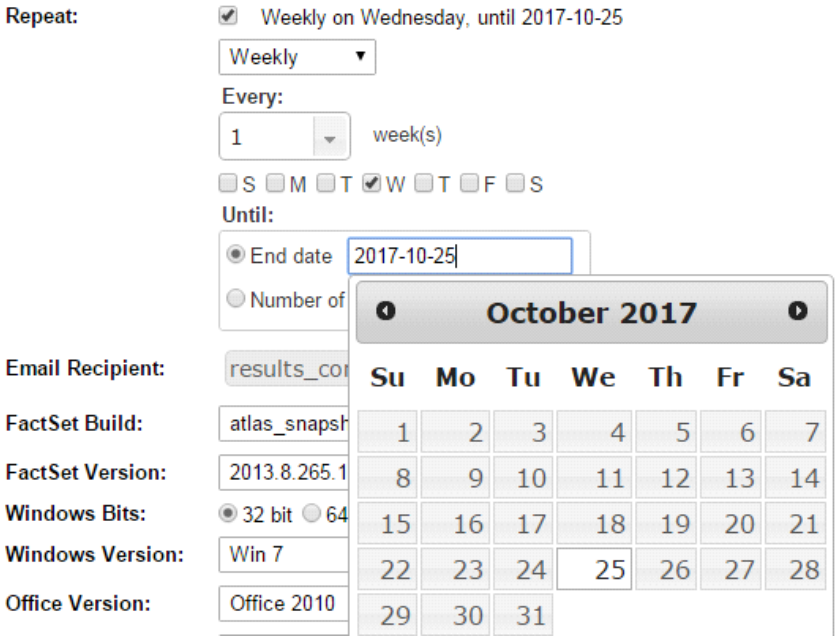
- Now for Recurrence, we choose the day of the week the QA lead requested the test plan to be scheduled.
- 99% of the case they want the jobs to run on Wednesday against default config and on Friday against Backwards Compatibility (there will be a separate job for each config)
- In our example, nearest Wednesday is Sept-9, and StartTime will be the time we calculated in the Spreadsheet.



- We follow a simple rule to calculate the time slot for a new job - Add 20 minutes to the start time of previous job. Example:

Commitment of Traders	Online- 2013.9	FDSQAR_C	Devel	Devel	LargesetCaccess	18:00 ET Wed	Default
Commitment of Traders	Online- 2013.9	FDSQAR_C	Devel	Staging	LargesetCaccess	18:20 ET Wed	Backwards Compatibility

- Commitment of Traders-Default config runs at 18:00 ET, so Commitment of Traders-Backwards Compatibility config will run at 18:20 ET.
- Next up is to add recurrence to the job. To do so, Click on Repeat checkbox, select "Weekly", 1, and "W" for Wednesday, and end date will be "Last Wednesday of the month 2 years from NOW"



- Now we are done entering the data for the job, so we submit the form.
- A GroupID will be generated for the job, which will be the GroupID of the very first instance of the series.
- Now open up the db and update the isRegressionJob flag for the job:

```
update CalendarParent set IsRegressionJob=1 where ParentID in
(select ParentID from CalendarEvent where GroupID in (89340))
```

- Where 89340 is the GroupID of the newly submitted job.
- Some of the online jobs, specifically PA3 jobs, need to run also on Tuesdays against previous week's SWP. To mark those as regression jobs we use isLateRegressionJob flag:

```
update CalendarParent set IsLateRegressionJob=1 where ParentID in
(select ParentID from CalendarEvent where GroupID in (87296))
```

- We also track the GroupIDs of the online jobs in a text file. So that we can reschedule the jobs just in case anything goes wrong.
- This concludes the process of adding a new test plan to Online Cron Schedule.

So as we saw, the manual parts that we do are:

1. Add the job details to the Spreadsheet
2. Schedule the job
3. Set isRegressionJob or isLateRegressionJob flags in the db
4. Store the GroupID in a text file

In AutoCron perspective, out of these steps,
Step-1 and 4 will not be required.
Step-2 will be taken care by QA leads instead of us.
Step-3 is the only thing that we need to set automatically.

So what I claim is we already have an implementation of AutoCron (excluding the db flags).

The simplest way to automate this process is:

1. Whenever a new cron job is submitted, if it has a recurrence of any of the kind - Build Creation or Days of week, we set isRegressionJob or isLateRegressionJob by either placing a hidden checkbox in the UI or by setting it programmatically by detecting the recurrence.
2. The db event instances will be generated and STAF CRON events will be registered as usual. - no changes here
3. SWP-GREEN takes care of updating the jobs with latest swp id once a trigger is received each week. - no changes here
4. One enhancement to swp-green will be to update the db with new swp id while updating the html files. So that the job data between db and html for a specific GroupID stay in sync.

For Build Creation option, as the user won't be able to provide the date when the first instance of the job will run, we can calculate the date ourselves in the UI itself and submit the job.

The current JEP calculates start date for future instances of the job depending on the start date, recurrence type (weekly, monthly etc), day name checkboxes etc options chosen while scheduling the job.
We can enhance it to calculate next Build Creation date, typically next Wednesday.

I think by enhancing swp-green script to let it create db and cron instances, we will be re-implementing the exiting functionality, probably in a complex way.

PA3 Cron Schedule Example

Wednesday, September 09, 2015 1:40 PM

Test Plan: PA3 auto-accounts

Day of Week	Build	Automation Account	VMs Stage	HTML Reports	Caccess	Appx. Start Time	Need to update SWP	IsRegressionJob	IsLateRegressionJob
Wed	Online-2013.8 I	FDSQAR_C	Devel	Devel	LargesetCaccess	07:00 AM ET Wed	Yes - Devel Default	1	0
Fri	Online-2013.8 I	FDSQAR_C	QA	Staging	LargesetCaccess	12:35 AM ET Fri	Backwards Compatibility	1	0
Tue	Online-2013.8 I	FDSQAR_C	QA	Staging	LargesetCaccess	19:00 PM ET Tue	Default	0	1

(From Cron scheduling spreadsheet)

Rules to Note:

- 99% of the case they want the jobs to run on Wednesday against default config and on Friday against Backwards Compatibility (there will be a separate job for each config)
- Some of the online jobs, specifically PA3 jobs, need to run also on Tuesdays against previous week's SWP. To mark those as regression jobs we use isLateRegressionJob flag = 1

Reference

Current Workflow to add a new test plan to Online Regression