SWP-Green and UI changes for AutoCron

Friday, September 04, 2015 12:38 AM

We need a simple and reliable way to register cron instances for all Online cron jobs upon Build Creation. Following are different ways of implementing it.

tl;dr:
Option-1 is to use a template.html file to create new GroupID.html files from.
Option-2 is to store the data that we store in a GroupID.html in a new db table.

Option - 1:

- UI side:
  1. Tellus JEP already has functionality to store entered data in db and generate GroupID.html file
  2. For Build Creation Option, the same functionality can be reused and everything except Start Date and Start Time will be stored in the db.
  3. A new html file will be created but this would be a template.html file for a specific test plan instead of a general GroupID.html file
  4. Further, this template.html can be saved as UniqueID.html file and the UniqueID will be stored in the db in a new table or existing tables with new column
  5. All the UniqueID.html file will be stored in a predefined location in Tellus machine
  6. There will be new db flags (say isRegressionJob) to mark this UniqueID as job to be executed upon build creation. It will be set to 1 automatically when the user selected "Run upon Build Creation" in the new JEP

- Script:
  1. Will retrieve all the UniqueIDs from db that have isRegressionJob=1, along with job register specific parameters – Environment, Project Type etc
  2. StartDate will be set to current date and an available slot to run this job will be retrieved from Priority Scheduling algo. The retrieved time slot will be set to StartTime.
  3. The script will then create a new GroupID and ChunkID and create new GroupID.html file with data from UniqueID.html file
  4. It will update the GroupID.html file to set GroupID and ChunkID in it
  5. Cron instance will be registered passing GroupID.html file, and other required parameters to the Dispatcher specific to ProjectType and Environment.

Option - 2:

- UI Side:
  1. Once a user submits a job to run upon Build Creation, all the data that goes to CalendarEvent/Parent will be stored as usual, including GroupID and ChunkID(s), except Start Date and Time
  2. All the data that the goes to GroupID.html will be stored in a new db table.
  3. The columns of the new db table will be same as each parameter/table cell title in GroupID.html file
  4. The GroupID from CalendarEvent/Parent for the job, will be the foreign key in the new db table (will be called UniqueID) and the isRegressionJob flag will be part of new db table

- Script:
  1. Upon receiving the trigger, script will retrieve all the UniqueIDs with isRegressionJob=1 from the new db table.
  2. for each UniqueIDs
     a. load the Group.html data from new db table and process it and generate GroupID.html file
     b. modify the swp build id in the GroupID to set the new swp id received in the trigger
     c. Set StartDate to current date and StartTime the available time slot retrieved from Priority Schedule algo
     d. Register a cron job passing GroupID.html file and other required parameters to the Dispatcher specific to ProjectType and Environment

Upon giving a second thought on both the options, I think Option-2 would be better as we will not be storing a separate template.html file which we may need to regenerate when a user updates the job to, say, include new test cases
In option-2 we just need to update the db column with new data.

I will update this note with more information tomorrow.
@vasu please add if I missed anything.