

Conversion Tools - Feature #3337

Feature # 1511 (New): Reporting v3

Source code analytics features

09/22/2017 01:47 PM - Eric Faulhaber

Status:	New	Start date:	
Priority:	Normal	Due date:	
Assignee:		% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:		vendor_id:	GCD
billable:	No		
Description			
Related issues:			
Copied from Conversion Tools - Feature #1514: implement fully dynamic reports...			Closed

History

#1 - 09/22/2017 01:47 PM - Eric Faulhaber

- Copied from Feature #1514: implement fully dynamic reports (backed by the database) added

#2 - 09/22/2017 02:00 PM - Eric Faulhaber

A summary of new features/improvements, originally posted in [#1514-177](#):

- Integrate preprocessor information (i.e., *.pphints data) with source view, so user can see where included content originated
- Investigate changing default filter profile
 - Note: the concept of the default profile has fairly complex/deep implementation implications; the definition of this profile is hard-coded in the batch report run which creates the initial database. Making this changeable is not recommended.
 - As an alternative, we may just want to remember which profile a user had active in their last session and apply that when they log back in; however, this has its own performance implications, since applying a non-default profile causes all overview reports to be rebuilt, which is very expensive.
- Disable browser back button
- Remove the red background from the menu icons, replace it with something more neutral (or make it transparent).
- Improve error output for search and custom reports.
 - The error messages for invalid condition and multiplex expressions currently reported come directly from the TRPL engine and aren't particularly user friendly
 - Currently, we display only the most specific error message from the Java exception chain
 - Making TRPL error messages significantly more user-friendly may not be feasible in the short run, but perhaps we can at least provide more of the original error output to help with some context
- Support multiple user accounts in addition to the initial admin account
 - Requires an administration page in the web app for the admin user to
 - create a user
 - delete a user
 - reset a password
- Add on-demand facility to change a password (currently, it is required only once, when logging in with the default password)
- Automatic password aging
- Pluggable valid password policy (low priority)
- Review APIs to determine if administrative authority should be checked before performing
 - Can probably be implemented by augmenting existing WebApi annotation with security information

#3 - 01/30/2019 12:52 PM - Eric Faulhaber

We should simplify the use of the ReportDriver. Currently, it is too error prone.

When implementing reporting v3, I inherited from v2 the idiom of running the ReportDriver twice, first for the schema reports, then for the code reports. Each run takes a number of arguments, many of which should not be left up to the user. Some of the arguments (like the report rule-sets to be run) must be hard-coded values for report generation to work at all. Furthermore, we should not allow any variability in the order in which the report batches are run. Finally, if we can presume that a conversion project is set up in the prescribed way (./abl subdirectory and ./cfg/p2j.cfg.xml file), many of the remaining arguments can be hard-coded within the ReportDriver or be derived by driver's logic.

I think all we really need to know is the name of the application, and we can derive or hard-code the rest of the information needed. From a user's point of view, the ReportDriver would just be launched once, with minimal information to specify. This would greatly simplify the Ant setup in build.xml and build.properties, reducing support effort.

We should not use any default value (like hotel) for the application name. It should default to empty string and if not overridden in the build.properties file, this should be detected by the ReportDriver, which should raise an error immediately. This last point probably can be applied to most settings in build.properties (not just those used for reporting purposes). We've seen that the failure to change a default value to a project-specific one has the potential to cause a harder to diagnose problem downstream. Better to know that a project-specific setting was overlooked as soon as possible.