

Conversion Tools - Bug #3336

Feature # 1511 (New): Reporting v3

Source code analytics defect punch list

09/22/2017 01:43 PM - Eric Faulhaber

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

History

#1 - 09/22/2017 01:45 PM - Eric Faulhaber

A summary of the remaining, known defects, originally posted in [#1514-177](#):

- Metadata tables/fields not represented in schema reports and source view
- Source view highlight not right for some schema artifacts (ranges not correct for tables and fields in many cases)
- Source view highlight not correct (2 lines off vertically) when viewing very bottom of any source file
- Source/AST view layout not correct when linked from Visualize Call Graph screen (extends beyond containing border horizontally)
- Old height algorithm used when resizing call graph view (new algorithm used for initial view) -- perhaps we add a skip list or selector parameter to getAvailableHeight function in main.js, so this algorithm can be used when components are already added to DOM.
- Loading overlay is stilted/blocked when used with a long-running client-side operation (e.g., when loading filter profiles for large projects), due to single threading on client
- AST view is very closely packed vertically and is illegible for broad trees (e.g., when viewing level 0 for most external procedures). Should extend beyond top and bottom view boundaries, requiring panning or zooming.
- Multi-user use is problematic in some cases
 - The active_file global temp-table is used in read mode for nearly every report; it is written to whenever a user logs in. If a long-running report is using this table in one session, a user can be blocked from logging in at all, since the request to acquire the write lock on this table will time out.
 - Activating a filter profile also causes writes to the shared active_file temporary table. So, though I have not tested this, I assume we will have similar issue when trying to activate a filter profile in one session while a long-running report is being built in another.
- Clicking on the expand/collapse "twisty" in the tabulator can be tricky. It seems so sensitive as to be error prone.
- Performance issues
 - Reports for large projects can take a very long time to build, especially with non-default file filter profiles
 - Ad-hoc search and custom reports can take a very long time to run with large projects -- they are single-threaded and visit every AST (except when active filter is applied for search)
 - Loading filter profiles causes a long-running client operation for large projects, as it leverages the checkbox tree's logic to manage nested branches/nodes
- Investigate: certain server APIs write to the database (e.g., adding a custom report, updating search history, etc.). Do we need to explicitly synchronize these APIs or is the database's normal locking behavior enough?

#2 - 09/22/2017 01:45 PM - Eric Faulhaber

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