Conversion Tools - Feature #5000

create a UI for managing the entire process/lifecyce of a conversion project

11/05/2020 10:34 AM - Greg Shah

Status:	New	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:			
billable:	No	**version:**	
vendor_id:	GCD		

Description

Related issues:

Related to Runtime Infrastructure - Feature #1845: create a simple tool (and ...	**New**

History

#1 - 11/05/2020 10:47 AM - Greg Shah

Our current approach has no UI. We just manually edit some set of text files, properties files, XML configuration files, templates and scripts to set the project up. This is error prone and difficult. We can't expect real customers to do this themselves when it is hard for us to do ourselves.

Providing a UI is a big part of the solution. The other part is eliminating as much of the unnecessary configuration/setup as possible.

The UI part would have:

- create/edit/delete a project
- add/remove/change files (4GL code and schema)
- control all possible configuration and hints
- helper wizards for complex tasks
- management of the micro-lifecycle of:
  - parsing
  - the reporting/gap analysis process
  - the conversion run
- by micro-lifecycle we mean that it should guide the setup/execution AND when there are problems it should guide the resolution of those issues
- management of the macro-lifecycle of the conversion project, especially the ability to customize, integrate changes over time
- in a perfect world this would include a syntax aware editor and the ability to write 4GL code, convert and run it all from the same UI
- web based if possible so we can create a kind of hosted conversion project environment
- and more...

It should be a kind of conversion project IDE. Options:

- extend our current Report Server UI to become this IDE
  - it is totally under our control
  - already web based
  - but we have to invent everything ourselves
- we could write an Eclipse plugin
  - this could be able to take advantage of useful existing IDE features
  - it may be very difficult to reuse our existing tooling
  - it is likely that our parsing/AST approach would need massive rework to fit into the Eclipse approach
  - we would live with the limits imposed by that technology, which are probably substantial

#2 - 11/05/2020 10:56 AM - Hynek Cihlar

One more benefit of using Eclipse is the environment is to some extent standardized and well known. This could attract more end users.

And another note. While it would be a lot of fun to write a full-featured IDE, doing so will require a LOT of effort. Even compared to the potential rework of our ASTs.

#3 - 12/28/2020 08:30 PM - Greg Shah

- Related to Feature #1845: create a simple tool (and docs) to provide easy creation/import of the configuration for a new server added