

Conversion Tools - Feature #4105

improve flexibility of schema loading

05/30/2019 10:31 AM - Greg Shah

| | | | |
|--|-----------------|------------------------|-----------|
| Status: | Closed | Start date: | |
| Priority: | Normal | Due date: | |
| Assignee: | Ovidiu Maxiniuc | % Done: | 100% |
| Category: | | Estimated time: | 0.00 hour |
| Target version: | | vendor_id: | GCD |
| billable: | No | | |
| Description | | | |
| Related issues: | | | |
| Related to Conversion Tools - Bug #5135: replace -s, -f and -x options of Con... | | Closed | |
| Related to Conversion Tools - Feature #6202: allow .df files to exist in a pa... | | Closed | |
| Related to Conversion Tools - Feature #6255: simplify the namespace configura... | | WIP | |

History

#1 - 05/30/2019 10:38 AM - Greg Shah

The following items need attention:

- It should be possible to specify a different logical database name from the .df basename. This isn't possible today because of problems in schema reporting (it fails when creating the default "all files" profile at the end of schema reporting).
- It should be possible to drop any/all schematas that are loaded by default using hints. This is needed for programs or directories which have conflicting schema usage.
- It should be possible to have more than one schema with the same logical database name. It is assumed that in this case, only one of those could be marked as load by default. Other than this, there should be no restriction.
- Schema configuration in p2j.cfg.xml should be much simpler. It can be as simple as just listing <namespace name="logical_db_name" />. Doing this should calculate default values for the .df filename, the .dict filename, whether it is loaded by default, which dialects are emitted and so on. When needed, these other things can be specified to handle the more rare cases.

#2 - 05/30/2019 10:48 AM - Eric Faulhaber

Greg Shah wrote:

It should be possible to specify a different logical database name from the .df basename. This isn't possible today because of problems in schema reporting (it fails when creating the default "all files" profile at the end of schema reporting).

To clarify this point, the problem we hit recently in this regard was an assumption in the build.xml that the .df and the .dict files had the same root name. This is not necessarily the case. So, the input to the schema reporting was wrong (the name of a .dict file that did not exist was passed as input to report generation). The error in the default "include all" profile processing at the end of schema reporting was a side effect of this error. That being said, the report generation should be fixed to warn or error out immediately on being passed bad file names, rather than failing by this unexpected side effect at the end.

#3 - 05/30/2019 11:53 AM - Eric Faulhaber

I think we should bake the input file parameters into the report driver, so that they don't need to be passed in by an application-specific build.xml script. They don't change from project to project (all temp-table and permanent .dict files for schema reports; all external procedure files for code reports), so there is no reason to require this error-prone parameter passing to be worked out for every project. As long as the project configuration is correct, this can be determined automatically.

Also, the current report worker makes assumptions about being called in two passes, first for schema reports, next for code reports. This also is error-prone and thus the report driver should do this for us. This will allow the build.xml scripts for report generation to be much simpler.

#4 - 05/30/2019 12:05 PM - Greg Shah

That being said, the report generation should be fixed to warn or error out immediately on being passed bad file names, rather than failing by this unexpected side effect at the end.

If we do this, we should warn but don't fail. Just process the ones that do exist.

#5 - 05/30/2019 12:05 PM - Greg Shah

Eric Faulhaber wrote:

I think we should bake the input file parameters into the report driver, so that they don't need to be passed in by an application-specific build.xml script. They don't change from project to project (all temp-table and permanent .dict files for schema reports; all external procedure files for code reports), so there is no reason to require this error-prone parameter passing to be worked out for every project. As long as the project configuration is correct, this can be determined automatically.

Also, the current report worker makes assumptions about being called in two passes, first for schema reports, next for code reports. This also is error-prone and thus the report driver should do this for us. This will allow the build.xml scripts for report generation to be much simpler.

I agree, this is the best way.

#6 - 05/30/2019 12:06 PM - Greg Shah

all external procedure files for code reports

This should be **all external procedure files and classes** for code reports.

#7 - 05/30/2019 12:12 PM - Eric Faulhaber

Greg Shah wrote:

all external procedure files for code reports

This should be **all external procedure files and classes** for code reports.

Can you think of any file inputs that can not currently be determined from the project configuration alone?

#8 - 05/30/2019 12:27 PM - Greg Shah

I think we should add a feature to the AST registry to provide us the 2 lists (all schema ASTs and all code ASTs). An AST cannot exist without being in the registry. It gets cleared when we ant clean a project and repopulated when we parse. It is the correct place for that logic. We don't want to embed the listing in the ReportDriver itself.

#9 - 09/02/2021 01:10 AM - Eric Faulhaber

- Assignee set to Ovidiu Maxiniuc

#10 - 09/02/2021 09:06 AM - Ovidiu Maxiniuc

- Status changed from New to WIP

Some clarifications, please:

- It should be possible to specify a different logical database name from the .df basename. This isn't possible today because of problems in schema reporting (it fails when creating the default "all files" profile at the end of schema reporting).

This means adding an optional attribute to namespace tag in p2j.cfg.xml and use it to identify the logical database instead of using the name? I think this is already supported, although not tested. The configuration will look like this:

```
<namespace name="app-logical-name"
  importFile="data/export-name.df"
  xmlFile="data/namespace/export-name.dict" >
```

- It should be possible to drop any/all schematas that are loaded by default using hints. This is needed for programs or directories which have conflicting schema usage.

Not sure what this means. An example would be useful, please.

- It should be possible to have more than one schema with the same logical database name. It is assumed that in this case, only one of those could be marked as load by default. Other than this, there should be no restriction.

I understand this like grouping multiple namespaces under a single <logical-db> node in p2j.cfg.xml. From this POV and schema conversion there is no issue, but I do not understand how will handle the code conversion the possible different tables/field types. I assume the default will be used and assume the other will simply match?

- Schema configuration in p2j.cfg.xml should be much simpler. It can be as simple as just listing <namespace name="logical_db_name" />. Doing this should calculate default values for the .df filename, the .dict filename, whether it is loaded by default, which dialects are emitted and so on. When needed, these other things can be specified to handle the more rare cases.

This is simple, I actually have already done it.

I do not have any questions for simplification of build.xml yet.

#11 - 09/02/2021 09:43 AM - Greg Shah

This means adding an optional attribute to namespace tag in p2j.cfg.xml and use it to identify the logical database instead of using the name? I think this is already supported, although not tested.

I believe that I tried this and it did not work for some reason. I think there is a "hidden" dependency somewhere.

Also, we should consider making the importFile and xmlFile optional. If not specified, they should default to the obvious values using the logical database name.

- It should be possible to drop any/all schematas that are loaded by default using hints. This is needed for programs or directories which have conflicting schema usage.

Not sure what this means. An example would be useful, please.

We have seen projects where most of the 4GL code uses one or more schemata as default=true while having a small amount of code (maybe just a few files or one sub-directory) which does not use those default schemata at all.

Consider a project in this structure:

```
module_one/  
module_two/  
module_three/  
module_four/
```

...
module_fifteen/

Let's say that all the code in this project used 3 schemata (important.df, awesome.df and essential.df) by default, except for module_nine/* and files module_three/some_program.p and module_four/whatever*.p. This means that module_nine/* and files module_three/some_program.p and module_four/whatever*.p will FAIL to parse if they "see" the important.df, awesome.df and essential.df as databases. This can happen if these schemata conflict with other schemata used in the project.

Today, we would not be able to mark these 3 schemata (important.df, awesome.df and essential.df) as default=true because the exception programs/directory can't process with them "connected". Instead, we must add a hints file for every directory EXCEPT for module_three, module_four and module_nine. In those modules that can have the connected default databases, the directory.hints files would include database entries for all 3 "default" schemata (important.df, awesome.df and essential.df). This is annoying but not too bad. Where it gets really messy is in module_three, module_four and module_nine where we can't use directory.hints and must instead encode a hints file for every single source program. Some would get the database entries for all 3 "default" schemata (important.df, awesome.df and essential.df) and others would get the custom schemata that they need.

If we could "drop" the default databases in hints files, then we could set important.df, awesome.df and essential.df as default=true in p2j.cfg.xml and only add hints for the module_nine/* (could have a directory.hints) and files module_three/some_program.p and module_four/whatever*.p which would have per-file hints. The amount of complexity would be much reduced and it would be easier to encode and maintain.

- It should be possible to have more than one schema with the same logical database name. It is assumed that in this case, only one of those could be marked as load by default. Other than this, there should be no restriction.

I understand this like grouping multiple namespaces under a single <logical-db> node in p2j.cfg.xml. From this POV and schema conversion there is no issue, but I do not understand how will handle the code conversion the possible different tables/field types. I assume the default will be used and assume the other will simply match?

We need to discuss this further. I admit I don't have the answers on this. See #4212 for the customer-specific discussion.

#13 - 09/07/2021 04:08 PM - Ovidiu Maxiniuc

Committed revision 12898.

It includes:

- The importFile, xmlFile attributes of namespace of p2j.cfg.xml are now optional and computed automatically if missing to default values. However, it is possible to overwrite these values if needed by using the old attributes. The namespace tag can be as simple as:
<namespace name="schema_name" />.
- Added new, much simplified syntax to ReportDriver which allows the file list parameters to be automatically detected from the configuration file. More than that, the new syntax will automatically compute and execute both passes in a single process. The process can be started without any parameters at all, in which case all files are automatically detected. The old syntaxes were kept. They can be useful if a partial report is needed, for various reasons. Here is the new ant target for hotel_gui project:

```
<!-- generate all reports -->
<target name="rpt-new"
  depends="init, init-ant-contrib"
  description="Generate code and schema reports from the already parsed project.">
  <java classname="com.goldencode.p2j.report.ReportDriver"
    fork="true"
    failonerror="true"
    dir="${basedir}" >
    <jvmarg value="-server"/>
    <jvmarg value="-Xmx4G"/>
    <jvmarg value="-XX:-OmitStackTraceInFastThrow"/>
    <jvmarg value="-DP2J_HOME=${p2j.home}"/>
    <jvmarg value="-DappAdminPass=${appname}"/>
    <classpath refid="compile.classpath"/>
  </java>
</target>
```

It will do the both passes. Optionally, an
<arg value="-Dn"/>
can be inserted if the debug level needs to be specified.

#14 - 09/08/2021 12:41 PM - Eric Faulhaber

Code review 3821c/12898:

This update contains changes for three issues: [#4105](#), [#5625](#), and [#4766](#). The [#4105](#) changes seem good to me. The other changes should be

reviewed in their respective tasks.

#15 - 09/08/2021 07:48 PM - Ovidiu Maxiniuc

- % Done changed from 0 to 60

Starting with r12902/3821c, it is possible to specify a different logical database name from the .df basename.

I even used a single schema file (.df) to create two different logical databases. It converted fine and there were no issues running the report (except for picking up some old .ast files from previous conversions).

#16 - 09/09/2021 02:18 PM - Greg Shah

Code Review Task branch 3821c Revision 12898

This is just for the ReportDriver changes. I don't expect the current code to change right now, but these 2 notes will be handled in some future improvement.

1. Any project that uses skeletons will not be able to use the new syntax. The reason: the skeletons have .ast files but they **must not** be included in reports since they are really not in the project. I recently added -X (ignore) mode to deal with this problem. The proper solution is to have the list of project files that were parsed as a recorded list and then drive reporting from that.

2. I don't like the idea that we are hard coding the ReportDriver to 4GL-specific code and data reports. And we are importing the schema package now. The changes make this code more specific to the Progress 4GL language and less able to be used as a general TRPL facility.

#17 - 09/13/2021 03:42 PM - Ovidiu Maxiniuc

Greg Shah wrote:

Code Review Task branch 3821c Revision 12898

1. Any project that uses skeletons will not be able to use the new syntax. The reason: the skeletons have .ast files but they **must not** be included in reports since they are really not in the project. I recently added -X (ignore) mode to deal with this problem. The proper solution is to have the list of project files that were parsed as a recorded list and then drive reporting from that.

I tried to fix this problem by using the registry.xml as you suggested in note-8 above. I added optional flags capability to stored AST nodes which allows now ReportDriver (and possible other utilities if we even need this) to query the list of processed files based on flags read from AST registry.

2. I don't like the idea that we are hard coding the ReportDriver to 4GL-specific code and data reports. And we are importing the schema package now. The changes make this code more specific to the Progress 4GL language and less able to be used as a general TRPL facility.

The schema package was an unused import. Dropped.

Committed revision 12923.

Since the AST flags in registry.xml are set at conversion time, the projects must be first reconverted before running the new report on them.

Please review.

#18 - 09/14/2021 09:30 AM - Greg Shah

Code Review Task Branch 3821c Revision 12923

The changes are very nice. I like the idea of being able to mark ASTs with custom flags and get lists of the ASTs that match. Great idea!

My only feedback is that the FLAG_DICT should not be defined in AstManager. The AstManager is a generic class, it is not even specific to TRPL and it definitely isn't specific to FWD. The FLAG_DICT flag should only exist in FWD, not even in TRPL itself.

#19 - 09/14/2021 10:06 AM - Ovidiu Maxiniuc

Greg Shah wrote:

My only feedback is that the FLAG_DICT should not be defined in AstManager. The AstManager is a generic class, it is not even specific to TRPL and it definitely isn't specific to FWD. The FLAG_DICT flag should only exist in FWD, not even in TRPL itself.

I thought the same. The reason I've chosen this location is because the flags must be 'reserved' somehow. Since the AST trees can be accessed from other places, some programmer may chose some overlapping flags which will lead to unexpected and most likely incorrect results.

#20 - 09/14/2021 10:28 AM - Greg Shah

I think that is OK, because the flags only need to be consistent within a given project. In other words, as long as the flags are all consistent within the FWD usage, then this will be OK.

#21 - 09/14/2021 10:29 AM - Greg Shah

I think the correct location is in AstGenerator (the same place we have the DICT_POSTFIX constant).

#22 - 09/14/2021 05:12 PM - Ovidiu Maxiniuc

Moved FLAG_DICT declaration from AstManager to AstGenerator. Augmented flags to 64 bits (8 reserved).

Committed revision 12927.

#23 - 09/17/2021 06:29 PM - Ovidiu Maxiniuc

The previous commit was pushing the metadata processing (standard.dict) to ReportDriver. Starting with revision 12946, the metadata schemata is excepted from being analysed by reporting tool.

I prepared the necessary changes for customer projects (including the hotel test projects) to take advance of the new syntax of ReportDriver. The old targets will be kept, for the moment, just in case a partial report is desired. I intentionally delayed this phase in order to have let all developers have full conversion of the project with the new flags in registry.

I am actively working on 2nd bullet. I did think a lot on 3rd one, but it seems undoable with the current approach to generation of DMOs. They are located in a package with a fixed name, that is the db logical name. If multiple logical databases share the same schemata, the information is duplicated. This is not optimal, but it works at this moment.

#24 - 09/18/2021 10:16 AM - Greg Shah

To clarify item 3:

- We have a customer which has 2 builds/compiles of their application:
 - A version that runs with a single schema named x and a single database instance of x. Let's call this the "merged" schema.
 - A version that runs with multiple schemata including "subset" x, a, b and c. The separate schemata are just the "merged" x split into multiple pieces, where one of the pieces is still called x. In other words, "merged" x = "subset" x + a + b + c.
- The "merged" x is never used at the same time as the "multiple schemata" x + a + b + c. The entire application source code is compiled with **either** the "merged" x or with the "multiple schemata" x + a + b + c.
- We will run conversion multiple times (twice in the example above).
- We just need to be able to configure these in the same project at the same time AND then select which ones to use for different conversion runs in the same project.
- The conversion output will never be used at the same time/together. The idea is to let there be multiple compiled versions of the converted code, each one will have a completely different set of DMOs.
- We don't need more than 1 logical database of the same name to be supported **at runtime** or even at conversion time, but we do need to be able to configure them and use them in different conversion runs.

#25 - 09/18/2021 10:16 AM - Greg Shah

We should not have to "swap out" the p2j.cfg.xml for different conversion runs.

#26 - 09/20/2021 06:59 PM - Ovidiu Maxiniuc

Since we need to run the conversion multiple times but keep the same p2j.cfg.xml it means the only way to inform the conversion driver is via some command line arguments of ConversionDriver or some property definitions passed with -D to java process. This parameter/property will choose which of the possible multiple configuration profiles stored in p2j.cfg.xml will be active at a given run of the conversion.

I chosen System property solution because it can be queried directly in the Configuration classes, without adding extra code for validation and management of the information. Also, the <schema> tag was split in multiple <profile>-s. There is always a reserved profile (unnamed) which will make things work as before. The test configuration contains the following:

```
[...]
<schema>
  <profile name="merged" default="true">
    <namespace name="x"
      default="true"
      importFile="data/x-merged.df"
      xmlFile="data/namespace/x-merged.df">
    </namespace>

    <namespace name="standard" />
    <!-- importFile="data/standard.df"-->
    <!-- xmlFile="data/namespace/standard.dict" />-->
  </profile>

  <profile name="split">
    <namespace name="x" />
    <!-- importFile="data/x.df"-->
    <!-- xmlFile="data/namespace/x.dict" />-->
    <namespace name="a" />
    <!-- importFile="data/a.df"-->
    <!-- xmlFile="data/namespace/a.dict" />-->
    <namespace name="b" />
    <!-- importFile="data/b.df"-->
    <!-- xmlFile="data/namespace/b.dict" />-->
    <namespace name="c" />
    <!-- importFile="data/c.df"-->
    <!-- xmlFile="data/namespace/c.dict" />-->
  </profile>
</schema>
```

```
<namespace name="standard" />
  <!-- importFile="data/standard.df"-->
  <!-- xmlFile="data/namespace/standard.dict" />-->
</profile>

<metadata name="standard">
  <table name="_Area" />
  [...]
```

Namespaces with same name can be define in multiple profiles, but they can have different configurations. In this case, the namespace x uses the x-merged.df data definition in merged profile, and the default x.df in the split profile. The standard profile must be defined in **all** profiles, but the metadata is defined only once, globally. By default, the merged profile is processed, but if `-Dfwd.profile.default=split` is added to conversion process as Java property definition, the second profile is used instead.

I am going to commit tomorrow, after some tests to make sure the "unnamed" profile loads correctly, to avoid any interference with current projects.

#27 - 09/21/2021 05:56 AM - Greg Shah

Overall, I like the approach very much.

I chosen System property solution because it can be queried directly in the Configuration classes, without adding extra code for validation and management of the information.

I understand this is convenient (avoids work on command line parameters), but:

- It is not consistent with how we configure the project. Users will not expect to use `-D` properties and it will be a source of confusion.
- It won't work with any code that calls the backing Java classes directly, which is something that is going to be important as we build out tooling for IDEs.

Let's please use a command line option approach.

FYI, I can see that we may want other configuration values stored in a "profile". This very likely will be a feature that is expanded in the future.

#28 - 09/24/2021 06:23 PM - Ovidiu Maxiniuc

The support for database profiles in configuration file was committed to 12996 of 3821c.

The active profile can be specified differently for each entry-point (I tried to maintain compatibility with old syntax as much as possible):

conversion: C<profile-name>
server: -profile <profile-name>
report: C<profile-name>
import: -profile <profile-name>

If nothing specific was configured all of these will continue to work as before.

Please review.

#29 - 09/29/2021 10:29 AM - Greg Shah

Code Review Task Branch 3821c Revision 12996

This is very nice! Some minor feedback:

1. As far as I know, there is no support for a P option in the conversion/transform/progress drivers, but the documentation has it listed. I think that is confusing.
2. Eventually (i.e. NOT NOW) the profile is going to be generic (not specific to database or schema usage). For example, we will be able to define different parameters like opsys in these profiles where today we are only able to define these globally and via directory/file hints. The reasoning is the same as for this schema usage: to create a single configuration file that can define related sets of configuration that can be used to convert the same 4GL code in different ways. For now, all that I want is to re-word the references to "database profile" or "schema profile" in our generic TRPL javadoc (PatternEngine and ReportDriver).

#30 - 09/29/2021 10:29 AM - Greg Shah

- % Done changed from 60 to 80

If I understand correctly, only the 2nd bullet remains. Correct?

#31 - 09/29/2021 12:18 PM - Ovidiu Maxiniuc

Greg Shah wrote:

Code Review Task Branch 3821c Revision 12996

1. As far as I know, there is no support for a P option in the conversion/transform/progress drivers, but the documentation has it listed. I think that is confusing.

It is documented as the default option. If -P is present or none of other file specification parameters the PLAIN mode is used (when the files to be processed are passed as parameter).

Note that -P, -S, -F, and -X are all incompatible but the syntax does not enforce them to be mutually exclusive. Also there is no validation for this so the user might pass invalid set of parameters, which will lead to possible non-deterministic and possible unexpected results.

Note that stating with this release, it is possible to add multiple options using - prefix. For example, we used to start the conversion with:

```
java -DP2J_HOME=<home> ConversionDriver -SID2T5 f2+m0+cb ./abl/ *.[wp]
```

using a single - option. This did not allow adding the profile except if it was a number eventually, like: -SID2T5P1 (where 2P1@ refer to the 1st profile). It is too ambiguous.

Now you can (and I think should be the preferred way) to write:

```
java -DP2J_HOME=<home> ConversionDriver -Cmerged -S -I -D2 -T5 f2+m0+cb ./abl/ *.[wp]
```

Even more, I think we should unify the entry points of all FWD applications with a common set of options (if possible) and using a more unified syntax. However, this requires all scripts to be changed and also the manuals, too. This is the reason I preferred to keep the syntax backward compatible.

2. Eventually (i.e. NOT NOW) the profile is going to be generic (not specific to database or schema usage). For example, we will be able to define different parameters like opsys in these profiles where today we are only able to define these globally and via directory/file hints. The reasoning is the same as for this schema usage: to create a single configuration file that can define related sets of configuration that can be used to convert the same 4GL code in different ways. For now, all that I want is to re-word the references to "database profile" or "schema profile" in our generic TRPL javadoc (PatternEngine and ReportDriver).

The way the configuration is parsed right now is that the profile can have only namespace -s as children, not the parameter-s. They can be added, of course, but the top-level ones also kept and used for default values. A property could be re-defined, this way overwriting the respective value for the current profile. It makes sense for the unnamed profile to not have any property (no property sub-nodes in schema) and inherit all from top-level.

If I understand correctly, only the 2nd bullet remains. Correct?

That is correct.

#32 - 09/29/2021 12:49 PM - Greg Shah

It is documented as the default option. If -P is present or none of other file specification parameters the PLAIN mode is used

I saw that in the documentation. But there is no actual processing for -P. Future developers will think there is something wrong because there is a documented option that actually has no backing code. I think that is more confusing than helpful. I'm not suggesting that we add processing for -P. I prefer to have no mention of a -P.

Note that stating with this release, it is possible to add multiple options using - prefix.

I saw that. And I like it.

Even more, I think we should unify the entry points of all FWD applications with a common set of options (if possible) and using a more unified syntax. However, this requires all scripts to be changed and also the manuals, too. This is the reason I preferred to keep the syntax backward compatible.

Agreed.

The way the configuration is parsed right now is that the profile can have only namespace -s as children, not the parameter-s.

Yes. I'm talking about future development here. For now, I just want some of the docs and syntax output to be cleaned up to remove the database/schema specific references.

#33 - 10/25/2021 10:49 AM - Ovidiu Maxiniuc

- Related to Bug #5135: replace -s, -f and -x options of ConversionDriver with a single combined mode added

#34 - 11/04/2021 05:19 PM - Roger Borrello

I am getting a warning:

```
[java] Invalid 'fwd.profile.default' definition. Auto-detecting default profile.
```

```
[java] Default profile is: unnamed
```

I do see that it is innocuous. Is the corrective action to prevent the warning to surround the namespaces within the schema with the profile name="merged" default="true" markup, and include -Dfwd.profile.default=merged in the conversion commands in the build.xml?

#35 - 11/05/2021 12:21 PM - Ovidiu Maxiniuc

Roger, please do not do anything, yet.

I forgot to remove implementation of profile selection from fwd.profile.default runtime property, as discussed with Greg (note 27). I will do it with my next commit and notify you.

#36 - 11/05/2021 12:29 PM - Roger Borrello

Ovidiu Maxiniuc wrote:

Roger, please do not do anything, yet.

I forgot to remove implementation of profile selection from fwd.profile.default runtime property, as discussed with Greg (note 27). I will do it with my next commit and notify you.

Thanks, Ovidiu. I will monitor for updates.

#37 - 11/08/2021 06:35 PM - Ovidiu Maxiniuc

Roger, please update to r13146.

The profile selection events are now properly logged.

#38 - 11/08/2021 07:31 PM - Roger Borrello

Ovidiu Maxiniuc wrote:

Roger, please update to r13146.

The profile selection events are now properly logged.

What config changes are there? Add the profile to the p2j.cfg.xml? Is there a command line option?

#39 - 11/08/2021 07:46 PM - Ovidiu Maxiniuc

Roger Borrello wrote:

What config changes are there?

These are related to profile selection/configuration. Do you use multiple profiles in your application?

Add the profile to the p2j.cfg.xml?

Yes, the profiles must be declared in p2j.cfg.xml.

Is there a command line option?

Yes, all entry FWD points have new options to select the active configuration profile. For example, the ServerDriver uses -profile. Unfortunately, the different syntaxes prevented the usage of same option across all applications. The ConversionDriver uses -P<profile> instead.

In the absence of such option set, FWD will choose one profile from p2j.cfg.xml, if there are many. The recent changes dropped the solution of defining the profile using a system property and uses proper logging API to log its selection instead of verbosly printing to System.out.

Note: for the moment, the profile only supports the schema definition, but other values will be added.

#40 - 11/08/2021 09:57 PM - Roger Borrello

Ovidiu Maxiniuc wrote:

Roger Borrello wrote:

What config changes are there?

These are related to profile selection/configuration. Do you use multiple profiles in your application?

We will in the future, but for now there is only 1. All the databases have been merged into 1, and there is a second just used for conversion name resolution. Later, we'll have to figure out how to have the multi-database setup configured.

Add the profile to the p2j.cfg.xml?

Yes, the profiles must be declared in p2j.cfg.xml.

This is what I had previous to this revision. The menu is the main database, and the records is used only during conversion:

```
<profile name="merged" default="true">
  <namespace
    name="standard"
    importFile="data/standard.df"
    xmlFile="data/namespace/standard.dict" />

  <namespace
    name="just_for_structure_no_data"
    importFile="data/just_for_structure_no_data.df"
    xmlFile="data/namespace/just_for_structure_no_data.dict"
    default="false" >
    <parameter name="ddl-dialects" value="postgresql" />
  </namespace>
  <namespace
    name="primary_merged_database"
    importFile="data/primary_merged_database.df"
    xmlFile="data/namespace/primary_merged_database.dict"
    default="true" >
    <parameter name="ddl-dialects" value="postgresql" />
  </namespace>
</profile>
```

Is that still a good configuration to use, and the profile is named "merged"?

Is there a command line option?

Yes, all entry FWD points have new options to select the active configuration profile. For example, the ServerDriver uses -profile. Unfortunately, the different syntaxes prevented the usage of same option across all applications. The ConversionDriver uses -P<profile> instead.

In the absence of such option set, FWD will choose one profile from p2j.cfg.xml, if there are many. The recent changes dropped the solution of defining the profile using a system property and uses proper logging API to log its selection instead of verbosely printing to System.out.

Note: for the moment, the profile only supports the schema definition, but other values will be added.

So the options are different for conversion versus starting the server? Would the below be correct for my convert.ignorelist target?

```
<!-- convert 4GL code from ${app.4gl.src} into ${src}, using specs and an ignorelist, in file-ignore-list.txt -->
<target name="convert.ignorelist"
  depends="init, check-file-ignore-list"
  description="Convert the 4GL source code to Java source code, using specs and a file-ignore-list.txt as input."
  if="file-ignore-list">
  <record name="cvt_${LOG_STAMP}.log" action="start"/>
  <java classname="com.goldencode.p2j.convert.ConversionDriver"
    fork="true"
    failonerror="true"
    dir="${basedir}">
    <jvmarg value="-server"/>
    <jvmarg value="-Xmx${conversionHeap}"/>
    <jvmarg value="-XX:-OmitStackTraceInFastThrow"/>
    <jvmarg value="-DP2J_HOME=${p2j.home}"/>
    <jvmarg value="-Drules.tracing=true" />
    <arg value="-IXd2"/>
    <arg value="f2+m0+cb"/>
    <arg value="${app.4gl.src}"/>
    <arg value="${app.4gl.pattern}"/>
```



```
<arg value="file-ignore-list.txt"/>
<arg value="-Pmerged"/>          <<<<<<<<<< new parameter
<classpath refid="compile.classpath"/>
</java>
<record name="cvt_${LOG_STAMP}.log" action="stop"/>
</target>
```

And finally, update server.sh to include -profile merged somewhere in the line that launches the server?

```
echo $prog $hprof $maxheap $srvr $dtxt $spi $cpath -Djava.system.class.loader=com.goldencode.p2j.classloader.M
ultiClassLoader com.goldencode.p2j.main.ServerDriver $mode $batch $cfg $port "$@"
```

#41 - 11/09/2021 06:47 AM - Greg Shah

Roger: Some elements can be removed from the namespace section because of improvements in this task. See item 4 in [#4105-1](#). That feature is already implemented.

#42 - 11/09/2021 03:17 PM - Ovidiu Maxiniuc

Roger Borrello wrote:

We will in the future, but for now there is only 1 [profile]. All the databases have been merged into 1, and there is a second just used for conversion name resolution. Later, we'll have to figure out how to have the multi-database setup configured.

Using different profiles (i.e. tables in different namespaces) leads the DMOs to be found in different Java packages, at least. The generated code must be run with the exact profile it was converted, otherwise the Java class resolution will probably fail.

This is what I had previous to this revision. The menu is the main database, and the records is used only during conversion:

[...]

Is that still a good configuration to use, and the profile is named "merged"?

The name is not important. Since you defined the merged profile with default attribute set, this is the one that will be used in the absence of any other option.

So the options are different for conversion versus starting the server?

Only the semantic for defining them. As noted above, I did not want to break the backward compatibility with existing scripts for conversion/import/server. Each of these have a different way of parsing the parameters, with different constraints like the order and their number. I tried to update both the output of the manual/help and the javadoc of the entry-point (main method) to reflect the latest syntax.

Would the below be correct for my convert.ignorelist target?

[...]

Yes. But I have a couple of notes:

- since there is already a default profile (or there is only one defined, or none, in which case one unnamed is automatically created) you are not required to add it in command line;
- the options to converter may now be passed independently. I think it is clearer than merging all of them in a single string. For example, the above arg set can be rewritten as:

```
<arg value="-I"/>
<arg value="-X"/>
<arg value="-D2"/>
<arg value="-Pmerged"/> <!-- new, optional parameter in this case -->
<arg value="f2+m0+cb"/>
<arg value="{app.4gl.src}"/>
<arg value="{app.4gl.pattern}"/>
<arg value="file-ignore-list.txt"/>
```

And finally, update server.sh to include -profile merged somewhere in the line that launches the server?

Yes! Something like this:

```
profile="-profile merged"
```

```
eval $prog -ea $hprof $maxheap $svr $dtxt $spi $cpath -Djava.system.class.loader=com.goldencode.p2j.classloader.MultiClassLoader
com.goldencode.p2j.main.ServerDriver $mode $profile $batch $cfg $port "$@"
```

#43 - 11/09/2021 03:53 PM - Roger Borrello

Ovidiu Maxiniuc wrote:

Using different profiles (i.e. tables in different namespaces) leads the DMOs to be found in different Java packages, at least. The generated code must be run with the exact profile it was converted, otherwise the Java class resolution will probably fail.

In this customer's environment, there will be separate conversions for the "merged" versus the "separate" databases.

The name is not important. Since you defined the merged profile with default attribute set, this is the one that will be used in the absence of any other option.

OK. I do prefer to name things explicit, especially if there will be multiple implementations.

So the options are different for conversion versus starting the server?

Only the semantic for defining them. As noted above, I did not want to break the backward compatibility with existing scripts for conversion/import/server. Each of these have a different way of parsing the parameters, with different constraints like the order and their number. I tried to update both the output of the manual/help and the javadoc of the entry-point (main method) to reflect the latest syntax.

- since there is already a default profile (or there is only one defined, or none, in which case one unnamed is automatically created) you are not required to add it in command line;
- the options to converter may now be passed independently. I think it is clearer than merging all of them in a single string. For example, the above arg set can be rewritten as:
[...]

Thanks for the tip. I'll cleanup the build.xml

And finally, update server.sh to include -profile merged somewhere in the line that launches the server?

Yes! Something like this:

```
profile="-profile merged"
```

```
eval $prog -ea $hprof $maxheap $srvr $dtx $spi $cpath -Djava.system.class.loader=com.goldencode.p2j.classloader.MultiClassLoader  
com.goldencode.p2j.main.ServerDriver $mode $profile $batch $cfg $port "$@"
```

I updated the p2j.cfg.xml and server.sh and received the below in server.log:

```
[11/09/2021 15:42:39 EST] (com.goldencode.p2j.schema.SchemaConfig:WARNING) Failed to set configuration profile  
(merged). Will use auto-detection.
```

So I assume this has to be updated in coordination with a conversion using the -P option?

Incidentally, Greg, I updated the p2j.cfg.xml to simplify per your direction. It now looks like:

```
<profile name="merged" default="true">  
  <namespace name="standard" />  
  
  <namespace  
    name="secondary_db"  
    default="false" >  
    <parameter name="ddl-dialects" value="postgresql" />  
  </namespace>  
  <namespace  
    name="main_db"  
    default="true" >  
    <parameter name="ddl-dialects" value="postgresql" />  
  </namespace>  
</profile>
```

Maybe there's more, but this is much cleaner!

#44 - 11/09/2021 03:57 PM - Greg Shah

Maybe there's more, but this is much cleaner!

Nice!

#45 - 11/09/2021 05:36 PM - Ovidiu Maxiniuc

Roger Borrello wrote:

I updated the p2j.cfg.xml and server.sh and received the below in server.log:

```
[11/09/2021 15:42:39 EST] (com.goldencode.p2j.schema.SchemaConfig:WARNING) Failed to set configuration pro  
file (merged). Will use auto-detection.
```

So I assume this has to be updated in coordination with a conversion using the -P option?

Roger,

I discovered recently that the configuration processing may be affected by the current classloader. As result, p2j.cfg.xml might have not been processed at all when attempted to be loaded in runtime mode. Please apply this patch:

```
--- src/com/goldencode/p2j/cfg/Configuration.java  
+++ src/com/goldencode/p2j/cfg/Configuration.java  
@@ -780,12 +780,19 @@  
  
+         if (isRuntimeConfig())  
+         {  
+             // load from jar  
+             ClassLoader cl = ClassLoader.getSystemClassLoader();  
+             cfgStream = cl.getResourceAsStream(CFG_DIR + "/" + DEF_CFG_FILE);  
  
-         if (cfgStream == null)  
-         {  
+             return config;  
+             // if the configuration file was found in jar, try locating it as normal file  
+             File cfgFile = new File(CFG_DIR + "/" + DEF_CFG_FILE);  
+             if (!cfgFile.exists())  
+             {  
+                 return config;  
+             }  
+             cfgStream = new FileInputStream(cfgFile);  
+         }  
+     }  
+     else
```

And let me know if this fixes the issue.

If not, please send me/post the p2j.cfg.xml and server.sh you used when you got this warning.

#46 - 11/11/2021 11:11 AM - Roger Borrello

Ovidiu Maxiniuc wrote:

I discovered recently that the configuration processing may be affected by the current classloader. As result, p2j.cfg.xml might have not been processed at all when attempted to be loaded in runtime mode. Please apply this patch:

[...]

And let me know if this fixes the issue.

If not, please send me/post the p2j.cfg.xml and server.sh you used when you got this warning.

I applied the patch, rebuilt FWD, and the error no longer appears in the server.log

#47 - 11/11/2021 11:21 AM - Ovidiu Maxiniuc

The patch was committed to 3821c in rev.13151.

#48 - 11/18/2021 11:27 AM - Greg Shah

Ovidiu: Did you ever finish making a plan to address item 3 in [#4105-1](#)? We need that last feature to implement #4212, which is needed by the end of 2021.

#49 - 11/18/2021 09:42 PM - Ovidiu Maxiniuc

Greg Shah wrote:

Ovidiu: Did you ever finish making a plan to address item 3 in [#4105-1](#)? We need that last feature to implement #4212, which is needed by the end of 2021.

I put some effort in this area today. I created a set of "tabu" schemas which are ignored when processing the namespaces. I am pleased with the result of the output. For the moment the set was manually altered by me for each processed file. The only thing remaining is to do the management of this set automatically, based on the hints I am going to implement next.

#50 - 11/19/2021 07:29 AM - Greg Shah

I like the idea.

#51 - 11/19/2021 08:08 PM - Ovidiu Maxiniuc

I could not finish the management of banned schemas. Things are a bit complicated because the SchemaDictionary\$Scope s and their Namespace s are reused from file to file (as an optimization, because these are time-consuming objects to build). Storing the set of unloaded schemas here will cause it to leak to next file processed.

I will think of a solution to avoid this. Of course, there is always the solution to use a static object, but I rejected the idea from the start because it will not work with parallel/multithread processing of the initial parsing of 4GL source files.

#52 - 11/22/2021 05:54 PM - Roger Borrello

I had tried to add Ovidiu as a watcher to #4212 so I could pose a configuration question in there, but I can ask here with generic terms. It's actual more of a review of my plan to handle 2 different database profiles for the same project.

There is a tradeoff between making my project artifacts contain all the information required for 2 different database profiles, and how complicated it becomes. I am trying to reach a balance.

In my p2j.cfg.xml, at the time I was only dealing with a single, merged database, I had greatly simplified by taking advantage of the advancements in this task:

```
<namespace
  name="standard"
  importFile="data/standard.df"
  xmlFile="data/namespace/standard.dict" />

<namespace
  name="secondary"
  importFile="data/secondary.df"
  xmlFile="data/namespace/secondary.dict"
  default="false" >
  <parameter name="ddl-dialects" value="postgresql" />
</namespace>
<namespace
  name="primary"
  importFile="data/primary.df"
  xmlFile="data/namespace/primary.dict"
  default="true" >
  <parameter name="ddl-dialects" value="postgresql" />
</namespace>
```

became

```
<profile name="merged" default="true">
  <namespace name="standard" />

  <namespace
    name="secondary"
    default="false" >
    <parameter name="ddl-dialects" value="postgresql" />
  </namespace>
  <namespace
    name="primary"
    importFile="data/merged/primary.df"
    xmlFile="data/namespace/merged/primary.dict"
    default="true" >
    <parameter name="ddl-dialects" value="postgresql" />
  </namespace>
</profile>
```

But when I introduced the "multi" database configuration, which breaks up the primary into a handful of separate databases, I judged it was better to reintroduce some directory structure back into the p2j.cfg.xml in order to be able to contain both sets of artifacts within the same project. This is because the secondary.df is standard between the profiles, and the primary.df is specific to both the merged and multi profiles. The directory structure is:

```
data/secondary.df
data/merged/primary.df
data/multi/primary.df
```

data/multi/primary_part1.df
data/multi/primary_part2.df

There are a few more, but it's enough to illustrate. So now I have this in p2j.cfg.xml:

```
<profile name="merged" default="true">
  <namespace name="standard" />

  <namespace
    name="secondary"
    default="false" >
    <parameter name="ddl-dialects" value="postgresql" />
  </namespace>
  <namespace
    name="primary"
    importFile="data/merged/primary.df"
    xmlFile="data/namespace/merged/primary.dict"
    default="true" >
    <parameter name="ddl-dialects" value="postgresql" />
  </namespace>
</profile>
```

```
<profile name="multi" default="false">
  <namespace name="standard" />

  <namespace
    name="secondary"
    default="false" >
    <parameter name="ddl-dialects" value="postgresql" />
  </namespace>
  <namespace
    name="primary"
    importFile="data/multi/primary.df"
    xmlFile="data/namespace/multi/primary.dict"
    default="true" >
    <parameter name="ddl-dialects" value="postgresql" />
  </namespace>
  <namespace
    name="primary_part1"
    importFile="data/multi/primary_part1.df"
    xmlFile="data/namespace/multi/primary_part1.dict"
    default="false" >
    <parameter name="ddl-dialects" value="postgresql" />
  </namespace>
  <namespace
    name="primary_part2"
    importFile="data/multi/primary_part2.df"
    xmlFile="data/namespace/multi/primary_part2.dict"
    default="false" >
    <parameter name="ddl-dialects" value="postgresql" />
  </namespace>
</profile>
```

With this, I can pass `-Dapp.db.profile=multi` on my ant command, and override the build.xml default (which is merged) since it contains:

```
<property name="app.db.profile" value="merged" />
```

The **ConversionDriver** commands contain `<arg value="-P${app.db.profile}" />`

When I perform the conversions, I use different directories at the root, but I can check out exactly the same project, just twice:

- `bzr co ~/secure/code/project_root project_abc`
- `bzr co ~/secure/code/project_root project_abc_multi`

The complexities of this particular project include:

- There is a different file-cvt-ignore.txt for one versus the other

- There is a different directory.hints to use for one versus the other

For the file-cvt-ignore.txt, right now I manually copied over the file-cvt-ignore.txt_multi to file-cvt-ignore.txt, but I should be able to just add a new target to the build.xml or modify the existing to handle a different filename within the target. I am working on that.

As for the directory.hints, it does appear I need specialized versions for the 2 profiles (not to mention I had 2 different sets of hints files at the top level).

One of them was:

```
<hints>
  <database name="secondary" />
</hints>
```

But now (conversion is in progress) I have:

```
<hints>
  <database name="secondary" />
  <database name="primary_part1" />
  <database name="primary_part2" />
</hints>
```

And the other wasn't necessary previously, but now I have (some non-relevant information removed):

```
<hints>
  <alias name="primary_part1" database="primary" />
  <alias name="primary_part2" database="primary" />
```

Since the conversion isn't complete, I am not sure if that is actually correct. I was using a previous version, so it might still break in conversion when I hit the directory that uses that hints file.

Any feedback as to whether this is a good approach or not would be appreciated. We did hear there is actually a 3rd profile that is required, where an additional primary_part2.df should be included for a specific set of customers.

#53 - 11/22/2021 05:57 PM - Roger Borrello

Made minor edit to the directory names, which were wrong.

#54 - 11/22/2021 06:30 PM - Ovidiu Maxiniuc

Roger,
The truth is: the project is quite complex.

For the different file lists, I think you can create one for each profile and name them file-cvt-ignore-multi.txt and file-cvt-ignore-merged.txt respectively. They will be referred in build.xml as file-cvt-ignore- $\$(app.db.profile)$.txt.

I do not have a solution for the directory hints. AFAIK, their name is fixed.

#55 - 11/22/2021 06:40 PM - Greg Shah

I do not have a solution for the directory hints. AFAIK, their name is fixed.

We have no solution for this. For now, you will need to swap them in and out.

#56 - 11/22/2021 06:51 PM - Ovidiu Maxiniuc

I have just committed revision 13194.

It contains support for 'unloading' a schema/database for a source-file or directory using hints. I tried to keep the syntax it as simple as possible. To the existing database element which has name attribute I added a new attribute load which have the default value true / yes. If set to false / no, elements from that schema will not be recognized (ignored) by the parser.

Short example: I have two schemas (fwd and fwd2), both with a permanent table named pt1 but fwd has and extra pt2 and fwd2 an extra table pt3. Consider following files:

```
// [fwd] schema MUST be disabled for this directory
```

```
FIND FIRST pt1 EXCLUSIVE-LOCK NO-ERROR.  
FIND FIRST pt3 EXCLUSIVE-LOCK NO-ERROR.  
FIND FIRST fwd2.pt1 EXCLUSIVE-LOCK NO-ERROR.  
FIND FIRST fwd2.pt3 EXCLUSIVE-LOCK NO-ERROR.
```

```
// [fwd2] schema MUST be disabled for this file
```

```
FIND FIRST pt1 EXCLUSIVE-LOCK NO-ERROR.  
FIND FIRST pt2 EXCLUSIVE-LOCK NO-ERROR.  
FIND FIRST fwd.pt1 EXCLUSIVE-LOCK NO-ERROR.  
FIND FIRST fwd.pt2 EXCLUSIVE-LOCK NO-ERROR.
```

and

```
// no conflicts
```

```
FIND FIRST fwd.pt1 EXCLUSIVE-LOCK NO-ERROR.  
FIND FIRST fwd2.pt1 EXCLUSIVE-LOCK NO-ERROR.
```

```
FIND FIRST pt2 EXCLUSIVE-LOCK NO-ERROR.  
FIND FIRST pt3 EXCLUSIVE-LOCK NO-ERROR.
```

The first two need the following hints:

```
<hints>  
  <database name="fwd" load="false"/>  
<!-- this will prevent loading the 'fwd' schema for associated source file -->  
</hints>
```

and

```
<hints>  
  <database name="fwd2" load="false"/>  
<!-- this will prevent loading the 'fwd2' schema for associated source file -->  
</hints>
```

respectively, either as specific file hints or in a parent directory.

The last source file does not need any hints since all the table names are well determined. Au contraire, unloading any of those schemas will cause this source file to fail to be parsed because it accesses tables from both db schemas.

#57 - 11/23/2021 08:19 AM - Greg Shah

- % Done changed from 80 to 100

- Status changed from WIP to Review

Code Review Task branch 3821c Revision 13194

It looks good.

Eric: Please review.

#58 - 11/23/2021 08:48 AM - Roger Borrello

- % Done changed from 100 to 80

Greg Shah wrote:

I do not have a solution for the directory hints. AFAIK, their name is fixed.

We have no solution for this. For now, you will need to swap them in and out.

Thanks. I suppose with a little bit of ant, a target could do that swap for me.

Regarding the configuration changes and my testing... the primary.p2o file was not created when I was performing the **merged** configuration. This had not been an issue until I moved the primary.df file into a merged directory. Is the **p2o** file handling able to utilize a subdirectory? Is there a tag like importFile or xmlFile to specify where it should go?

#59 - 11/23/2021 09:24 AM - Greg Shah

I think we may have some hard coded assumptions that everything is in the data/ directory. You should move away from the subdir idea.

#60 - 11/24/2021 08:16 AM - Roger Borrello

Greg Shah wrote:

I think we may have some hard coded assumptions that everything is in the data/ directory. You should move away from the subdir idea.

The problem with that is it runs counter to this task. If I created 2 p2j.cfg.xml files, one for my merged and one for my multi, what is the benefit of using profiles? Can "p2oFile" be another TAG_NAMESPACE like "importFile" and "xmlFile"?

#61 - 11/24/2021 08:32 AM - Greg Shah

Why do you need 2 p2j.cfg.xml files?

#62 - 11/24/2021 08:34 AM - Greg Shah

I don't understand why the .df files have to be in sub-directories. You are only laying down the .df files needed for a given build, so they can be in the same directory. Or you can name them different names and put them all in place, but only use some of them because a given schema is chosen at build time.

#63 - 11/24/2021 09:25 AM - Roger Borrello

Greg Shah wrote:

I don't understand why the .df files have to be in sub-directories. You are only laying down the .df files needed for a given build, so they can be in the same directory. Or you can name them different names and put them all in place, but only use some of them because a given schema is chosen at build time.

It just comes down to process. All the different .df files are collected at once from within their separate directories and placed in the same manner. I assumed I could drive the conversion via the configuration capabilities in p2j.cfg.xml by specifying where each of the different .df files are located.

It is a design decision about where to put the relocating of the .df files to match the p2j.cfg.xml given its limitations. Perhaps **build.prep.multi** and **build.prep.merged** targets in the build would do it?

As an aside, I did end up being able to use a single set of directory.hints files for the build (thanks for the default="true" hint), so it is coming down this for my task.

#64 - 11/25/2021 08:59 AM - Greg Shah

Just name the .df files differently (even if they have the same logical db names) and they can all be placed at once.

#65 - 12/06/2021 08:19 AM - Greg Shah

- Status changed from Review to Closed

- % Done changed from 80 to 100

#66 - 03/23/2022 08:38 AM - Greg Shah

- Related to Feature #6202: allow .df files to exist in a path that is not directly in data/ added

#67 - 04/06/2022 06:42 PM - Greg Shah

- Related to Feature #6255: simplify the namespace configuration, part deux added