Core Development - Feature #1474

Feature # 1434 (New): Conversion reference documentation and training slides

write Hibernate Mappings chapter of P2J Conversion Reference

08/09/2012 11:57 PM - Eric Faulhaber

Status:	Review	**Start date:**	08/09/2012
Priority:	Normal	**Due date:**	
Assignee:	Adrian Lungu	**% Done:**	90%
Category:		**Estimated time:**	0.00 hour
Target version:			
billable:	No	**vendor_id:**	GCD

Description

History

#1 - 08/10/2012 01:07 AM - Eric Faulhaber

The hibernate_mappings.odt document is a placeholder for a P2J Conversion Reference chapter about the purpose and content of the Hibernate Mapping documents ({DMOClassName}.hbm.xml) generated by the P2J schema conversion process. So far, it is empty.

This chapter should not re-hash the Hibernate documentation about HBM XML documents, though it should refer to the Hibernate reference documentation. Rather, the point of this chapter is to convey an understanding of the relationship between the schema constructs from Progress (i.e., from within the .df file for a permanent database, within DEFINE TEMP-TABLE statements for temp-tables) and the elements in the Hibernate mapping documents generated by P2J. To be clear, this should not be a detailed explanation of how the P2J schema conversion process works internally, but rather a reference of how the inputs from Progress map to the outputs created by conversion (however, you probably will need to dig into how the schema conversion process works internally in order to be able to write this content).

Source/background material for this chapter should include:

- the *Overview* chapter for Part 3 of the Conversion Reference (contains high level discussion and multiple mentions of the Hibernate mappings);
- the rule sets and templates in rules/schema (particularly hibernate.xml and hibernate_templates.tpl); these will probably lead you to more rule sets and into Java code;
- the training slides for the developer1 course (see schema_conversion.odp); should help to explain the process and give you clues as to which rule sets to review.

DO NOT use the information in the com.goldencode.p2j.schema package summary. This is way out of date and no longer reflects the P2J implementation.

The chapter should include an example of the inputs and outputs for both a permanent database table and a temp-table. The permanent table can re-use the example provided in the slides (though it may need to be fleshed out a bit -- the slide omits some content). The temp-table example should include normal fields (one for each major data type), plus a field with an extent value (i.e., a Progress array). The extent field example will generate an association in the Hibernate mapping document, since P2J normalizes Progress extent fields into separate, related database tables. The Hibernate mapping document manages this relationship.

#2 - 08/10/2012 01:12 AM - Adrian Lungu

- Status changed from New to WIP

#3 - 08/16/2012 01:21 PM - Adrian Lungu

- % Done changed from 0 to 60

First version committed into bazaar.
Topics covered:

- general Hibernate mapping file format
- table mappings (both permanent/temporary cases)
- fields mappings (both permanent/temporary cases)

There are some TODOs left in the document. I do not know if:
- the content should be structured on 1st level (permanent/temporary) and 2nd level (table/fields) or as it is now
- more examples are needed to cover all the field data types (for both permanent/temporary tables)
- more in depth explanations on each *.df* syntax element(and also *define temp-table*) are needed as this are covered in the schema_disposition.odt.

#4 - 08/21/2012 03:49 PM - Adrian Lungu

- Status changed from WIP to Review

- % Done changed from 60 to 90

- finished all content
- commited to bazaar

#5 - 08/24/2012 02:21 AM - Adrian Lungu

These are some comments on the final hibernate_mappings.odt version:

1. The examples are covering all data types with the exception of row-id data type.
2. Table 1 not updated to include the syntax for <one-to-one> <one-to-many> and <many-to-one> elements.
3. I assumed that for each many-to-one relation a corresponding one-to-many relation is also generated. I couldn't find a way to generate only one type.
4. There are no (complete)explanations in the *Hibernate Associations for Foreign Relations* section related to natural joins, but I included references to other sections.
   - maybe the *Natural Join Analysis* section from p2j/src/com/goldencode/p2j/schema/package.html could be used here.
5. In the case of one-to-many associations the DMO implementation has the following problem:
   in the generated file {DMO}Impl.java on this line:

```
    public Set getChildSet();
```

   the error is:

```
    error : "Set cannot be resolved to a type"
```

   solvable by:

```
    public java.util.Set getChildSet();
```

   or

```
    import java.util.Set; /* I think this is missing */
```