

Database - Feature #1582

add conversion and runtime support for sequences

10/11/2012 11:00 PM - Eric Faulhaber

Status:	Closed	Start date:	11/12/2012
Priority:	Normal	Due date:	
Assignee:	Ovidiu Maxiniuc	% Done:	100%
Category:		Estimated time:	0.00 hour
Target version:	Runtime Support for Server Features	version:	
billable:	No		
vendor_id:	GCD		
Description			
Subtasks:			
Feature # 1910: add schema conversion support for sequences			Closed
Feature # 1911: add runtime support for sequences			Closed
Bug # 2178: Asymmetric sequence behavior			New

History

#1 - 10/11/2012 11:14 PM - Eric Faulhaber

The following schema properties (with the noted settings) must be supported by schema conversion:

- initial value of 0, 1 or 100
- increment of 1
- cycle on limit of no and yes
- minimum value of 0, 1 or 100
- maximum value (it is only sometimes explicitly specified) of 9999, 999999999 or 2000000000

The following language constructs must be implemented:

- CURRENT-VALUE statement
- NEXT-VALUE() function
- CURRENT-VALUE() function

These statements/functions should be implemented with a public API that is emitted into converted business logic. That public API should be backed by a database-specific implementation if possible (i.e., the behavior of a native, database sequence can be made to match the 4GL sequence behavior exactly). Note that this will require sequence creation/setup during schema conversion. If a native sequence cannot be used, a default Java implementation should be provided by the P2J runtime. The decision of whether a native implementation or the default, Java implementation is used should be delegated to the P2J database dialect.

The same public API should also support the dynamic forms (DYNAMIC-CURRENT-VALUE statement, DYNAMIC-NEXT-VALUE(), DYNAMIC-CURRENT-VALUE()) with essentially no additional work. The syntax is the same, the features and behavior are the same and in Java the parameter handling naturally resolves the essential difference (which is that the non-dynamic forms have a database and/or sequence names statically defined versus using an expression at runtime).

#2 - 10/22/2012 02:13 PM - Eric Faulhaber

- Start date deleted (10/15/2012)

#3 - 10/31/2012 04:13 PM - Greg Shah

- Target version set to Milestone 7

#4 - 10/31/2012 04:14 PM - Greg Shah

- Target version changed from Milestone 7 to Milestone 3

#5 - 11/05/2012 11:29 PM - Eric Faulhaber

- Target version changed from Milestone 3 to Milestone 7

#6 - 11/15/2012 05:26 AM - Ovidiu Maxiniuc

At first attempt to read about and prepare the dynamic forms testcases I could not find them in Progress Reference document. After digging into the OE11 documentation CD, I found in ABL Reference manual the DYNAMIC forms. However, with this occasion I realized that in OE, they have have an extra optionally parameter (tenant-id). Moreover, the non-dynamic functions are also enriched with this parameter. I suppose this should be implemented in the new form, as OE reference describes.

#7 - 11/15/2012 07:53 AM - Ovidiu Maxiniuc

Seems like the OpenEdge version installed on windev01 is only at version 10.2 and does not support the new extra parameter from OE11 that I read in the documentation. I did not understand yet what is it good for, but I believe P2J implementation will focus on OE10.x at the moment so the P2J implementation will not support it.

#8 - 11/15/2012 09:50 AM - Eric Faulhaber

Ovidiu Maxiniuc wrote:

... I believe P2J implementation will focus on OE10.x at the moment so the P2J implementation will not support it.

Correct.

#9 - 11/22/2012 11:01 AM - Ovidiu Maxiniuc

- File *om_upd20121122a.zip* added

Uploaded update patch that covers both conversion and runtime sub-tasks (1910 and 1911).

#10 - 11/23/2012 01:03 AM - Eric Faulhaber

I have done only a very high-level review of the update so far, and it looks like you've addressed all the necessary areas, though it seems there is still some work to do to complete the implementation.

To begin, I have a few basic questions:

- It appears the sequence DDL is generated into a separate file from the main DDL. How hard would it be to append this DDL to the main DDL file? We separate the index DDL from the main DDL because the data import process creates the indexes after importing the data, but it seems sequence creation belongs with the main DDL. The simpler we can keep the process (i.e., fewer files to manage) for the DBAs which must set up the new databases, the better.
- It looks like there may be some classes in this update that are not related to this feature set (e.g., ConnectionManager). Please remove those.

- Is the version of progress.g what you and Greg finally decided upon? It looks like there is some work in process in there that has been commented out.
- Can you please help me understand the idea behind the Sequence class hierarchy? In particular, TemporarySequence is a bit of a mystery to me. Can you also explain the idea behind making Sequence a singleton? We may need to have multiple instances of different types available simultaneously. Consider an application that uses multiple databases from different vendors, where we cannot support one of the vendor's implementations with NativeSequence (i.e., we would need to implement an emulated sequence in the P2J runtime, presumably with a sibling class of NativeSequence?). This is probably not a common scenario, but our design should not preclude it.

In general, please document these classes more fully, providing detailed JavaDoc at the class, variable, and method level, including non-public members. Non-public methods should have all parameters documented. Please do not begin method names with underscores unless there is a very specific reason (i.e., not just because they are non-public).

A very good start overall. I will have more detailed feedback later.

#11 - 11/23/2012 12:04 PM - Ovidiu Maxiniuc

- File *om_upd20121123a.zip* added

My answers/comments to your questions:

- There is no separate ddl file generated for sequences. Only the xml that generates sqls is a new one, it will append its output at the end of sql tables.
- Sorry about ConnectionManager, it slipped in, it contains changes from a different issue, it is removed in today's patch.
- I believe the progress.g is the final one. I removed the comments as they are not needed any more as sequence statements are handled globally with similar syntax statements.
- Indeed, TemporarySequence is not finalized - it was in my intention to implement some kind of in-memory manager. I believe it will not reach the bzr repository. There was not in my intention to create a true singleton. From Constantin I understood that the project can be configured at database-level with the dialect/sql server. I implemented a CustomSequence that handles sequences inside the P2J code as you mentioned. It works even better than the native one on PostgreSQL and I manually tested the queries on H2 too (they are commented in source code).

I added some more comments. Yesterday version was not a final one, but more like an intermediate, to be sure I'm on the right track here. Neither is this version, but I believe I am getting closer.

#12 - 11/23/2012 01:26 PM - Greg Shah

- Status changed from *New* to *WIP*

I have reviewed the progress.g changes. They are fine. My only request is for you to please change the ending copyright date from 2011 to 2012.

#13 - 11/24/2012 03:31 PM - Eric Faulhaber

- Assignee set to *Ovidiu Maxiniuc*

#14 - 11/28/2012 11:25 PM - Eric Faulhaber

Some additional code review comments for the 1123a drop:

- How is CustomSequence to be used? AFAIK, Progress sequences are incremented with nextval whether or not the enclosing transaction commits or rolls back (this assumption needs to be confirmed/refuted with a testcase). If my assumption is true, you will have to make the updates to your backing table in a separate transaction scope (i.e., in a separate service thread). Please describe the way this will work.
- The lookup code in CustomSequence is duplicated in 3 places, is potentially inefficient for databases with many sequences, and may return the wrong answer if multiple databases have sequences with the same names. Why are you searching by name if the ldbName parameter is null in these cases? Shouldn't you consult with the superclass only (the assumption being that a null ldbName indicates the first connected database -- this assumption is indicated in the comments for Sequence.getDatabaseForSequence(String), btw, what is the basis for this assumption)?
- Why does the Sequence.getDatabaseForSequence(String) method take the sequence name as a parameter? The parameter isn't used. Is this method meant to be overridden by a subclass (currently, it is not)?
- language_statements.rules: cut and paste error: comment at line 74 still refers to substring statement.
- generate_seq_ddl.rules: IIRC, there is no need to cast assignments of constants into variables (see lines 61-64). It doesn't hurt, but the expression compiler will auto-box these assignments.
- P2JDialect (and subclasses): is there a need to define the new APIs with parameters that are Java wrapper classes, or can primitives be used here? We typically prefer primitives and avoid using the wrappers unless there is a need (e.g., we need to represent the lack of a value using null, or an upstream or downstream API needs the wrappers instead of primitives). If we use wrappers instead of primitives, the JavaDoc should explain why.

General format/other comments:

- Please omit the "JPRM" column from the header of all new source files. This is obsolete, but we keep it around in older files.
- Don't import individual classes in import statements unless resolving a conflict across packages. We prefer the package.* syntax.
- When wrapping a method signature onto multiple lines, please align the start of the second and subsequent lines with the start of the parameter type on the first line.
- For double-slash comments, please add a space between the double-slash and the first character of the comment.
- Please put all class and instance variables at the top of the class (e.g., sequences in Sequence.java); organize all static methods before instance methods.
- When chaining method calls across multiple lines, please align the dots so the chaining is obvious.

#15 - 08/12/2013 01:51 PM - Eric Faulhaber

- Status changed from WIP to Closed

#16 - 11/16/2016 11:42 AM - Greg Shah

- Target version changed from Milestone 7 to Runtime Support for Server Features

Files

om_upd20121122a.zip	357 KB	11/22/2012	Ovidiu Maxiniuc
om_upd20121123a.zip	343 KB	11/23/2012	Ovidiu Maxiniuc