Base Language - Feature #1641

implement SAX XML support

10/21/2012 11:51 AM - Greg Shah

Status:	Closed	**Start date:**	02/05/2013
Priority:	Normal	**Due date:**	06/14/2013
Assignee:		**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:	Runtime Support for Server Features		
billable:	No	**version:**	
vendor_id:	GCD		

Description

Subtasks:

Feature # 1987: conversion support for SAX	**Closed**
Feature # 1988: runtime support for SAX	**Closed**

History

#1 - 10/21/2012 11:51 AM - Greg Shah

Priority features:

CREATE-SAX-READER; CREATE-SAX-WRITER; METHODS: get-value-by-index(), start-element(), end-element(), write-characters(), set-input-source(), set-output-destination(), start-document(), declare-namespace(), end-document(), insert-attribute(), write-data-element(), write-cdata(), write-fragment(), sax-parse(), sax-parse-first(), sax-parse-next(), set-node(); ATTRIBUTES: LOCATOR-LINE-NUMBER, LOCATOR-COLUMN-NUMBER, PARSE-STATUS, FORMATTED, STRICT, HANDLER, ENCODING;

#2 - 10/31/2012 01:37 PM - Greg Shah

- Target version set to Milestone 7

#3 - 02/05/2013 07:58 PM - Greg Shah

Please document the full list of SAX-related features (e.g. everything related to the SAX-attributes/SAX-reader/SAX-writer object handles, and the CREATE SAX-ATTRIBUTES statement, CREATE SAX-READER statement and CREATE SAX-WRITER statement). The list above are the high priority features, but it would be good to get the conversion support and stubs written for everything. But if something is too difficult to get done now (and it is low priority) then we will defer it.

#4 - 02/06/2013 04:57 PM - Eugenie Lyzenko

SAX-READER, SAX-WRITER, SAX-ATTRIBUTES Objects:

The already mentioned statements, attributes and methods:

CREATE SAX-ATTRIBUTES
CREATE SAX-READER
CREATE SAX-WRITER

ENCODING
FORMATTED
HANDLER
LOCATOR-COLUMN-NUMBER
LOCATOR-LINE-NUMBER
PARSE-STATUS
STRICT

DECLARE-NAMESPACE
END-DOCUMENT
END-ELEMENT
GET-VALUE-BY-INDEX
INSERT-ATTRIBUTE
SAX-PARSE
SAX-PARSE-FIRST
SAX-PARSE-NEXT
SET-INPUT-SOURCE
SET-OUTPUT-DESTINATION
START-DOCUMENT
START-ELEMENT
WRITE-CDATA
WRITE-CHARACTERS
WRITE-DATA-ELEMENT
WRITE-FRAGMENT

Not yet mentioned attributes and methods:

ADM-DATA
FRAGMENT
HANDLE
INSTANTIATING-PROCEDURE
LOCATOR-PUBLIC-ID
LOCATOR-SYSTEM-ID
NONAMESPACE-SCHEMA-LOCATION
NUM-ITEMS
PRIVATE-DATA
SCHEMA-LOCATION
SCHEMA-PATH
STANDALONE
SUPPRESS-NAMESPACE-PROCESSING
TYPE
UNIQUE-ID
VALIDATION-ENABLED
VERSION
WRITE-STATUS

ADD-SCHEMA-LOCATION
COPY-SAX-ATTRIBUTES
GET-INDEX-BY-NAMESPACE-NAME
GET-INDEX-BY-QNAME
GET-LOCALNAME-BY-INDEX
GET-QNAME-BY-INDEX
GET-TYPE-BY-INDEX
GET-TYPE-BY-NAMESPACE-NAME
GET-TYPE-BY-QNAME
GET-URI-BY-INDEX
GET-VALUE-BY-NAMESPACE-NAME
GET-VALUE-BY-QNAME
REMOVE-ATTRIBUTE
RESET
STOP-PARSING
UPDATE-ATTRIBUE
WRITE-COMMENT
WRITE-EMPTY-ELEMENT
WRITE-ENTITY-REF
WRITE-EXTERNAL-DTD
WRITE-PROCESSING-INSTRUCTION

Greg, where can I find the SAX-READER object description? Do you have Progress 10.x pdf documents? May be my docs are outdated, I have scanned langref, epi, proghand... and have not found any reference for SAX-WRITER object.

#5 - 02/06/2013 05:45 PM - Greg Shah

From our document called "Getting Started with P2J Development":

```
1. Download the Progress 4GL documentation. There are 2 versions of interest:

Progress 4GL v9.1E: http://download.psdn.com/documentation/openedge/pdf_zip_files/91e/progress_documentation_9
.1e_pdfs.zip
Progress ABL v11: http://download.psdn.com/documentation/openedge/pdf_zip_files/110/oepdfs.zip.
```

#6 - 02/07/2013 11:16 AM - Eugenie Lyzenko

I have updated the SAX related attributes/methods list. The question for method set-node() you noted. I have not found this method for SAX related objects. May be you mean the SET-ATTRIBUTE-NODE method for DOM related handles?

#7 - 02/07/2013 11:21 AM - Greg Shah

Yes, you're right that SET-NODE is not for SAX. It is for SOAP support. Don't work on it here.

#8 - 02/07/2013 03:32 PM - Eugenie Lyzenko

Implementation plan:
1. The create statements will refer to static methods in XmlHelper class.
2. The methods and attributes will refer to newly created interfaces/classes: SaxAttributes, SaxAttributesImpl, SaxReader, SaxReaderImpl, SaxWriter, SaxWriterImpl and SaxEntity, SaxEntityImpl for common attributes/methods.
3. The interfaces will inherit WrappedResource and be added into handle unwrap methods.
4. In parallel to attribute/methods stubs implementation the testcase to check conversion will be created.

Is it OK approach? I have already made the changes in rule file to convert create statements.

#9 - 02/07/2013 03:40 PM - Greg Shah

Everything is good in the plan, except I don't want XmlHelper to be the factory for create statements. Just add them to the XmlFactory and add the rest of the Sax* classes to the same .../p2j/xml/ package.

#10 - 02/07/2013 03:46 PM - Eugenie Lyzenko

> Everything is good in the plan, except I don't want XmlHelper to be the factory for create statements. Just add them to the XmlFactory and add the rest of the Sax* classes to the same .../p2j/xml/ package.

Yes, of course. I just made mistake in description, so really I thought about XmlFactory, not XmlHelper which is really not the good place for such modifications.

#11 - 02/07/2013 04:14 PM - Greg Shah

Excellent, thanks!

#12 - 02/07/2013 04:58 PM - Eugenie Lyzenko

I've found the strange conflict.
1. We need to import .xml.* package to the converted file.
2. It is happening in methods_attribute.rules
3. But create sax-* statements are handled in language_statement.rules(the same was as for create DOM XML objects)
4. So assume the following case: we have Progress source when the only SAX related element is create sax-reader for example.

Yes, this is strange to create object and does not use it(neither by attribute access nor by method invocation) but if this happening - the converted source will not be compiled. Or this case will be considered as never used code and useless create statement will be removed from resulting converted code? Or this is impossible at all(because it is up to Progress programmer to eliminate such situations)? Consider the source file like this to illustrate:
---
def var hSaxReader as handle.
create sax-reader hSaxReader.

#13 - 02/07/2013 05:09 PM - Greg Shah

Just add similar import creation code to the language_statements.rules. It doesn't hurt if the JAST gets more than 1 of these. We will only emit it 1 time into the java source.

#14 - 02/08/2013 01:27 PM - Eugenie Lyzenko

One point to clarify:

In some methods I see the LONGCHAR as possible data type for parameters. It is not currently supported and I can just ignore this case?

#15 - 02/08/2013 01:57 PM - Greg Shah

Actually, you are testing the new longchar data type support right now (in staging). Please map any longchar stuff to a longchar type and anything that can be either character or longchar should now be coded to their common base class Text. You can see both new classes in staging right now.

#16 - 02/11/2013 04:49 PM - Eugenie Lyzenko

- File sax_xml_test.p.20130211.zip added

The testcases has been uploaded. All statements inside are converted and compiled with stubs. I'm preparing Javadoc enties for java sources.

Do we need the rest of the attributes/methods(~40) to be added now?

#17 - 02/11/2013 06:03 PM - Greg Shah

Yes, go ahead and add most of the others. Exclude the following attributes (they are not specific to SAX): ADM-DATA, PRIVATE-DATA, INSTANTIATING-PROCEDURE, HANDLE, TYPE.

#18 - 02/13/2013 08:45 AM - Eugenie Lyzenko

I have encountered one issue. The attribute
UNIQUE-ID
is common not only for SAX XML objects but for DOM XML too. So we need the common java class for such situations. Say XCommon.java for file source. The common features will be implemented in this class. Is it OK?

#19 - 02/13/2013 12:09 PM - Greg Shah

Actually, UNIQUE-ID is common to more than just the DOM and SAX support. Other resources (e.g. procedure handles, SOAP...) also use it.

Constantin: where do you think this is best located?

#20 - 02/13/2013 12:30 PM - Eugenie Lyzenko

The other common attributes/methods:

NONAMESPACE-SCHEMA-LOCATION attribute(common with X-Document object)
ADD-SCHEMA-LOCATION method(common with X-Document object)

#21 - 02/13/2013 12:39 PM - Greg Shah

Both of these are only used for SAX and DOM, so they can go in an XCommon interface.

#22 - 02/14/2013 06:02 PM - Greg Shah

How soon can you provide an update for these features?

#23 - 02/14/2013 06:15 PM - Eugenie Lyzenko

> How soon can you provide an update for these features?

In a hour or so. I need to sync with the latest code, make final checking and upload.

#24 - 02/14/2013 08:05 PM - Eugenie Lyzenko

- File sax_xml_test.p.20130214a.zip added

- File evl_upd20130114a.zip added

The SAX conversion and test file has been uploaded for you to review. Merged with recent P2J codebase.

The UniqueId is not fully supported until we determine where to implement it.

#25 - 02/15/2013 12:35 PM - Greg Shah

Code Review Feedback:

1. Change the Abstract and class Javadoc in XmlFactory and XEntity so that they mention both SAX and DOM.

2. In XEntityImpl:

```
   public boolean isUnknown()
   {
      return super.isUnknown() && (hXEntity == null || xnode == null);
   }
```

I think the logic should probably be an OR:

```
      return super.isUnknown() || (hXEntity == null || xnode == null);
```

If the super is unknown, wouldn't the XEntity always be unknown too?

3. Please use smaller parameter names in all of your classes. I think the result can be just as informative, but it could read better.

XCommon/XCommonImpl:
nonamespaceSchemaLocation -> location
targetNamespace -> namespace
attrName -> name
namespaceUri -> uri

SaxReader/SaxReaderImpl:
isSuppressNamespaceProcessing -> suppress
isValidationEnabled -> enable
schemaLocation -> location
schemaPath -> path

SaxEntity/SaxEntityImpl:
attributeName -> name
attributeValue -> value
namespaceUri -> uri

SaxWriter/SaxWriterImpl:
namespaceUri -> uri
saxAttrHandle -> handle

SaxAttributes/SaxAttributesImpl:
srcAttrHandle -> handle
attrName -> name
attrValue -> value

4. In SaxAttributes/SaxAttributesImpl, XCommon/XCommonImpl, there are some index-related methods that take int or integer. Please use double in place of int and NumberType in place of integer. This allows the result to work for the full range of possible numeric inputs.

Overall: this is a very good update, well done!

The plan from here:

1. Make the changes noted above.
2. Merge with the bzr check-ins that should happen from the updates being tested in staging.
3. Upload that result for a final code review.

If possible, I would like to get these things done today.

From there I will merge a few files with an update from CA and one from myself. I'll apply the changes to staging and reconvert. I'll do this tonight or tomorrow morning. Then you will run the harness as soon as possible after that.

#26 - 02/15/2013 01:21 PM - Eugenie Lyzenko

> ...
> 2. Merge with the bzr check-ins that should happen from the updates being tested in staging.
> ...

Does it means the changes I need to merge with are already in bzr? The testing is still in progress, not completed. Do I need to wait for harness to be done without regression to merge the update?

#27 - 02/15/2013 01:46 PM - Greg Shah

Normally, we only check in after everything is passed in regression testing. For today only, I have asked Eric to check in the current changes that are being tested. I will let you know as soon as bzr is ready.

#28 - 02/15/2013 02:00 PM - Eugenie Lyzenko

> Normally, we only check in after everything is passed in regression testing. For today only, I have asked Eric to check in the current changes that are being tested. I will let you know as soon as bzr is ready.

As an alternative plan: you just send me the update instead of checking in bzr. I'll merge the changes but repository remains untouched until harness is done.

#29 - 02/15/2013 02:06 PM - Greg Shah

I appreciate it, but there are 3 of us that need to merge and it will be easier to use bzr.

#30 - 02/15/2013 05:21 PM - Eugenie Lyzenko

- File evl_upd20130215a.zip added

The update merged with bzr 10173 has been uploaded.

#31 - 02/15/2013 05:47 PM - Greg Shah

It looks good. I will merge it on top of Constantin's update and upload the result here.

#32 - 02/15/2013 07:07 PM - Greg Shah

- File evl_upd20130215b.zip added

This is merged on top of ca_upd20130215d.zip (#1608) and ca_upd20130215e.zip (#1620 and #1621). It MUST be applied AFTER those updates. This will be applied to staging and tested shortly.

#33 - 02/18/2013 08:28 AM - Eugenie Lyzenko

evl_upd20130215b.zip has been committed as 10176.

#34 - 02/18/2013 09:05 AM - Greg Shah

I have not cross-referenced your final code with the list of all possible features. Can you please make a list of all features that are NOT yet implemented in the conversion AND/OR which do not yet have stubs in the runtime.

#35 - 02/18/2013 09:18 AM - Eugenie Lyzenko

The following attributes(SAX XML support) not yet implemented according to your note(a means SaxAttr, r means SaxReader, w means Sax Writer, (r) means readonly, (r/w) means writable).:
ADM-DATA ar(r/w)
HANDLE arw(r)
INSTANTIATING-PROCEDURE arw(r)
PRIVATE-DATA ar(r/w)
TYPE arw(r)

Also the following Attribute is in unclear state because it is used in many other places outside SAX and DOM:
UNIQUE-ID ar(r)(integer)

#36 - 03/19/2013 08:13 AM - Greg Shah

Eugenie, please implement the following:

These constants are recognized by the parser but we do not emit them as literals in the conversion.

SAX-UNINITIALIZED
SAX-RUNNING
SAX-COMPLETE
SAX-PARSER-ERROR
SAX-WRITE-IDLE
SAX-WRITE-BEGIN
SAX-WRITE-TAG
SAX-WRITE-ELEMENT
SAX-WRITE-CONTENT
SAX-WRITE-COMPLETE
SAX-WRITE-ERROR

Look in the parser (search for "\nliteral" by enabling regular expressions). In the javadoc, you will see the "well known" integer constants that map to these symbolic names. Please create corresponding constants in our runtime that match, for example:

/** SAX-UNINITIALIZED 4GL compiler constant. */
final static int SAX-UNINITIALIZED = 1;

I'm not sure which class is best for these, probably SaxEntity is best.

Then you will need to change convert/literals.rules to emit something like SaxEntity.SAX-UNINITIALIZED when you encounter KW_SAX_UNIN.

Please also review how these can be used. I want to make sure we have all the signatures needed to take the int constants if needed. My concern is that if we have a signature that can only take an integer type, then we need to add signatures or make the SaxEntity constant an integer type instead of int so that everything will compile.

The "literals" function in common-progress will also need to be enhanced for this to work.

#37 - 03/19/2013 08:18 AM - Constantin Asofiei

- File ca_upd20130319c.zip added

- File ca_upd20130319b.zip added

Eugenie: attached you have the runtime support for this; please go ahead with the conversion support.

Greg: the correct class for them IMO is SaxReader for PARSE-STATUS and SaxWriter for WRITE-STATUS constants.

The values set at the constants in SaxReader and SaxWriter are the ones used by 4GL.

#38 - 03/19/2013 09:45 AM - Greg Shah

Code Feedback (Eugenie: you can make the final changes to the code Constantin posted):

1. Please make the constants all public final static instead of just public (which is editable, something we don't want).

2. Add history entries.

3. Go ahead and apply the spelling fix ("atribute" to "attribute") in SaxWriterImpl.

#39 - 03/19/2013 11:08 AM - Constantin Asofiei

> 1. Please make the constants all public final static instead of just public (which is editable, something we don't want).

AFAIK any field declared in an interface is by default "static final" (the compiler takes care to put these modifiers if they are omitted).

> 3. Go ahead and apply the spelling fix ("atribute" to "attribute") in SaxWriterImpl.

It's best to do a replaceAll for "atribute" with "attribute" in the entire file.

#40 - 03/19/2013 11:41 AM - Greg Shah

> AFAIK any field declared in an interface is by default "static final" (the compiler takes care to put these modifiers if they are omitted).

Good point. But if the reader can easily misunderstand that (like I did), then I prefer to have it be explicitly defined.

#41 - 03/20/2013 02:36 PM - Eugenie Lyzenko

- File evl_upd20130320a.zip added

The update includes:
1. Changing for constant types.
2. Adding history entries.
3. Fixes for misspelled words in SaxWriterImpl and SaxReaderImpl.
4. Adding function "sax_literals" to detect the related literal constant.
5. The convert/literals.rules modified to handle SAX literals listed before. The sample file sax_literals.p now converted and compiled.

I'm starting complete conversion test on my local system for this update.

#42 - 03/20/2013 02:46 PM - Greg Shah

Code feedback: everything looks good.

The only issue is that you need to merge with the recent bzr revision. Please do that, repost the update and restart conversion testing.

#43 - 03/20/2013 03:11 PM - Eugenie Lyzenko

- File evl_upd20130320b.zip added

This update merged with bzr 10302. Starting the conversion testing.

#44 - 03/20/2013 04:40 PM - Greg Shah

If it passes testing you can check it in and distribute it.

#45 - 03/20/2013 05:46 PM - Eugenie Lyzenko

> If it passes testing you can check it in and distribute it.

Conversion testing has been passed. I'm going to commit.

#46 - 03/20/2013 05:54 PM - Eugenie Lyzenko

The update evl_upd20130320b.zip has been committed in bzr as 10303.

#47 - 04/25/2013 10:37 AM - Greg Shah

Comment from CA:

I suppose we will use something like Apache AXIS2 for the low-level communication? I don't think is needed to implement the protocol from scratch.

This and #2022 should be worked by the same person, as they are related (I think the web services need the SOAP features internally). And about

the X-Document features: serialized XMLs (built using X-Document, then saved to a longchar var) can be passed as input to SOAP service invocation. We should be able to build the SOAP/Web Services without prior X-Document features, but we should test them with the XMLs too.

#48 - 10/08/2013 05:37 AM - Greg Shah

When do you expect to have a first version for review?

#49 - 10/08/2013 07:19 PM - Evgeny Kiselev

- File evk_upd20130922a.zip added

Greg Shah wrote:

> When do you expect to have a first version for review?

First version for review will be available at the beginning or maybe in the middle next week

#50 - 10/17/2013 07:32 PM - Evgeny Kiselev

what means in documentation A CHARACTER or LONGCHAR expression evaluating to the value of the attribute. for example:

```
INSERT-ATTRIBUTE ( attribute-name , attribute-value [ , namespaceURI ] )
attribute-name
A CHARACTER or LONGCHAR expression evaluating to the fully qualified or
unqualified name of the attribute.
```

I've tried to set longchar into method, but 4gl throw a exception that parameter list invalid. Character type works fine.

#51 - 10/18/2013 04:44 AM - Constantin Asofiei

I think there is a good chance that the progress documentation is wrong; although the code passes, I couldn't find a way for the INSERT-ATTRIBUTE to work when it has longchar arguments. Maybe they didn't got a chance to finish the support? My testcase was this (I was thinking the codepage had something to do with this, but it didn't work):

```
def var h as handle.

create sax-attributes h.

def var lc1 as longchar.
def var lc2 as longchar.
fix-codepage(lc1) = "iso8859-1".
message get-codepage(lc1).

lc1 = "language".
lc2 = "en".

h:insert-attribute(lc1, lc1).
```

My opinion at this time is we should show the appropriate error message if the INSERT-ATTRIBUTES receives a longchar parameter.

#52 - 10/20/2013 09:17 PM - Evgeny Kiselev

- File evk_upd20131020a.zip added

First drop for review:
1) SaxAttributes is close to finish version, there are some not important TODOs
2) SaxEntityImpl will be abstract and clean some overloaded methods
3) SaxWriterImpl I'm still experimenting between event and stream style. And I think stream style is more applicable. This class is not fully implemented because need some more time for testing. Writer statuses is fully implemented with right error messages.
4) SaxReaderImpl validation is not implemented. Also I'm still working on SaxHandler and procedure calls.

#53 - 10/21/2013 03:51 AM - Constantin Asofiei

Evgeny, following is my review. The main concerns are about abstracting the sax reading and writing, to not write code specific to an input source or output destination type. Please take a look and let me know your opinion. The logic of the rest of the code looks good.

1. XCommonImpl

- please put isHandleTypeOf in handle class and make sure the javadoc states that the handle must be valid too.

2. SaxEntity

- please add javadoc to the Source(String) c'tor

3. SaxReader

- ReadStatus enum needs javadoc and instance field formatting fixed.
- are you sure SAX_UNINITIALIZED must have the "SAX_WRITE_IDLE" name?

4. SaxWriter

- WriteStatus enum needs java doc and fields formatting fixed
- when you have enums with more than one parameter, you can format them like this (same applies to ReadStatus):

```
      SAX_WRITE_IDLE    (1, "SAX-WRITE-IDLE"),
      SAX_WRITE_BEGIN   (2, "WRITE-BEGIN"),
      SAX_WRITE_TAG     (3, "SAX-WRITE-TAG"),
      SAX_WRITE_ELEMENT (4, "SAX-WRITE-ELEMENT"),
      SAX_WRITE_CONTENT (5, "SAX-WRITE-CONTENT"),
      SAX_WRITE_COMPLETE(6, "SAX-WRITE-COMPLETE"),
      SAX_WRITE_ERROR   (7, "SAX-WRITE-ERROR");
```

  It's easier to read the parameters if they are all column-aligned.
- line 150 - make sure the curly braces are on their own lines
- removeAttribute - please fix the javadoc formatting.

5. SaxEntityImpl

- there is no need to add constants for the error numbers

6. SaxReaderImpl

- setHandler - check if the hCallback parameter is validated to be a non-proxy procedure handle when the attribute is set. For some other resources I've worked on, considering that a procedure handle can be deleted between the assignment and its actually usage, I found that the procedure handle is validated when it is used, no when it is set. More, please don't save BDT references; the setHandler code should be this.hCallback = new handle(hCallback) or this.hCallback.assign(hCallback).
- in some javadoc you have progressice instead of progressive
- from official docs, I see that the during callbacks the SELF system handle is set to the SAX-READER instance. To do this, always bracket the callback with SelfManager.pushSelf and popSelf calls. SelfManager.self() can be used to determine the SAX-READER object during callback.
- some ideas for parsing: as you can use different input source types (i.e. file, variable, stream), I think you need to wrap the original input source using a custom input source, so that the backing source is hidden and the parsing can use this generic input source.
- when a callback is executed, a i.e. EndElement procedure is invoked from the handler procedure context; I think you need test that the procedure exists and parameters match before invoking it.

7. SaxWriterImpl

- same as for sax reading, I think you need to use an absract output source, so that the writing does no depend on the output destination type. More, when using i.e. longchar, does the longchar var hold intermediate results? If it does not, then it means the writing is done in a private in-mem location and when is finished, this data is flushed to the output destination. I think the same can be checked for reading - what happens if the i.e. longchar var set as input gets altered during reading? Does the reading stop or an error occurs?
- in setOutputDestination, I don't think you need to save the filename/etc. I think is better to build custom output streams depending on the destination and use these output streams during writing. Same for reading.

8. SaxAttributesImpl

- checkArguments can be added as BaseDataType.elementsOfType and its signature changed to:

```
public static boolean elementsOfType(BaseDataType... arr, Class<? extends BaseDataType> cls)
```

#54 - 10/30/2013 10:01 PM - Evgeny Kiselev

- File evk_upd20131030a.zip added

- Status changed from New to WIP

1) I have question about procedure handle:
- in handle I have ExternalProgramWrapper where I can get real instance of java class with specific procedure. But what is the best way todo next ? I've tried to call need methods via reflection. But I've find Callback and CallbackResolver classes but they can't invoke Methods.

2) In update I've fixed most of the issues from note 53.

#55 - 10/31/2013 03:39 AM - Constantin Asofiei

1. handle.isHandleTypeOf
   - please do not use an existing class name for variables/parameters. It is confusing. I'm referring to the handle handle parameter.
2. SaxReaderImpl.setInputSource

- for FILE source, you can't use Java IO to read the file, as the file might be on a different machine or on a relative location not available from the server JVM. Idea is, in 4GL, all IO resources are client-side. More, the docs state that the FILE can be an URL. You can use the StreamFactory.openFileStream APIs to obtain a P2J Stream instance, which you can then use to read the file. In case of URLs, we will need something different.
- for MEMPTR source (and others, including file): you assume that the XML is already generated and make a copy of it in a ByteArrayInputStream. But what if the XML is i.e. streamed via a socket and written to the MEMPTR as the data is coming? Idea is, please test what happens if the XML is not fully loaded in the specified source when set-input-source is called; just add some data to the source **AFTER** the parsing has started or after the set-input-source is called and see what happens. More, between the SAX-PARSE and the set-input-source calls, the input source might "disappear" completely: the file can be deleted, the stream closed, the memptr/longchar var changed/set to unknown. What happens in these cases? You need to check these cases before going forward.

The idea of SAX parsing via events AFAIK is that you don't even have to load the entire XML in memory, you process its input source sequentially; and from this, we can assume the entire XML might not be available upfront. To make myself clearer, I was expecting for you to create some InputStream implementations for each input source type; the implementation will call i.e. read on the input source when the parsing tells it to.

Lets focus on making the SAX-reader work first and after that move to SAX-writer. Once we have this working, we might be able to apply the same pattern to the SAX writing.

> 1) I have question about procedure handle:

To invoke an internal procedure for a procedure handle, I think is enough to call ConrolFlowOps.invokeInWithMode(<procedure name>, handler, <argument modes>, <argument 1>, <argument 2>, ..., <argument n>). But I don't see any answers about how the procedure handle is validated between the HANDLER assignment and its usage during events - have you tested this yet?

#56 - 11/05/2013 09:15 PM - Evgeny Kiselev

- File evk_upd20131105a.zip added

from note 55
1. Fixed
2. Added special wrapper for File/Stream/memptr/longchar. In 4gl method set-input-source is not important, first read will be always in methods parse or parse-next. But than parse is started, for example for memptr - all data should be already inside memptr.
3. Added initial version of callback handle ProcDefaultHandler class.

3. ConrolFlowOps.invokeIn - how I can specify output parameters ?

#57 - 11/06/2013 03:46 AM - Constantin Asofiei

> 2. Added special wrapper for File/Stream/memptr/longchar. In 4gl method set-input-source is not important, first read will be always in methods parse or parse-next. But than parse is started, for example for memptr - all data should be already inside memptr.

I don't see any validation of the source parameter in the set-input-source or additional validation of the input source while reading it. More, have you checked what I've asked in note 55, about what happens with the SAX parsing if the source gets modified after the parse has started? Please post the results here.

> 3. Added initial version of callback handle ProcDefaultHandler class.

1. if the class is only for internal usage, make it package private
2. you need to test what happens if a certain internal procedure (associated with a SAX parser notification) does not exist or its signature does not match (parameter modes, types, order, etc) - does the parsing end, is it ignored, something else? Although the procedure name and parameter validation might already be implemented by the ControlFlowOps.invoke APIs, you need to make sure it is the same.
3. to pass explicit parameter modes, use ConrolFlowOps.invokeInWithMode as I've stated in my previous note. The modes are a single-string with each character one of O, U, I for Output, Input-Output and Input modes; each character represents the mode of the corresponding parameter (by position). i.e.

```
ControlFlowOps.invokeInWithMode("NotationDecl", procHandle,
                                "III",
                                new character(name),
                                new character(publicId),
                                new character(systemId));
```

to invoke a procedure with all parameters as INPUT. Note that the parameter modes you pass here might not be the same as the procedure definition; in this case, an error is raised.
Please post here the case for which you need an OUTPUT parameter. Because for OUTPUT and INPUT-OUTPUT parameters, you need to pass the exact reference received from the application logic.

4. the throws declarations for the class methods should not be there. I think any parsing exception should be caught and i.e. an appropriate error message should be shown. Determine this by testing: what happens if an XML is malformed? Does the parsing stop? Is the error ignored?
5. the javadoc of the methods are just copied from the org.xml.sax.helpers.DefaultHandler; this is not enough, please change (and format it) it so that it explains what the methods do, 4GL-wise.

To conclude, the update is a step forward, but you need to make sure the edge cases and validation/error handling are found and treated correctly.

#58 - 11/15/2013 12:20 PM - Evgeny Kiselev

- File evk_upd20131115a.zip added

> I don't see any validation of the source parameter in the set-input-source or additional validation of the input source while reading it. More, have you checked what I've asked in note 55, about what happens with the SAX parsing if the source gets modified after the parse has started? Please post the results here.

1) I've added some validation here. Source can be modified in any time and if it's broke parser, it will throw IO exception (14586).
2) ProcDefaultHandler is specify in update, but I've found that java SAX API is not correct. Here like SAX-WRITER we need to use StAX api. Because either if in 4gl we call parse method, 4gl parse all document via sax-parse-next and sax-parse-first so it's like stream API. that's why changing input source is possible. SAX API doesn't give opportunity to parse tag by tag. So in ProcDefaultHandler need to review end of class.

> you need to test what happens if a certain internal procedure (associated with a SAX parser notification) does not exist or its signature does not match (parameter modes, types, order, etc) - does the parsing end, is it ignored, something else? Although the procedure name and parameter validation might already be implemented by the ControlFlowOps.invoke APIs, you need to make sure it is the same.

1) if callback is not exists, parse just skip invocation of procedure and just continue.
2) checking parameters do before callback invocation and throw error if parameters mismatched.
3) I've write my own(draft) implementation of parameters and procedure exist checking, because it's very hard to take this information form ConrolFlowOps. Most of needed methods is private is has deep call hierarchy and most of methods are throw exceptions. Maybe we need some refactoring if we want to use this code.

> the throws declarations for the class methods should not be there. I think any parsing exception should be caught and i.e. an appropriate error message should be shown. Determine this by testing: what happens if an XML is malformed? Does the parsing stop? Is the error ignored?

1) most of exception goes to callback handler (handler has error, fatalError, warning methods). If exception IO or some fatal error, than parsing is stoped and throw exception. In other case all exception goes to callbacks.

2) ProcDefaultHandler is specify in update, but I've found that java SAX API is not correct. Here like SAX-WRITER we need to use StAX api. Because either if in 4gl we call parse method, 4gl parse all document via sax-parse-next and sax-parse-first so it's like stream API. that's why changing input source is possible. SAX API doesn't give opportunity to parse tag by tag. So in ProcDefaultHandler need to review end of class.

I don't understand what you mean here. Please post an example.

ProcDefaultHandler.invokeProcedure:

- you need to check if the passed handle is a procedure before casting it (use ProcedureManager.isProcedure)
- is not OK to check internalEntries.indexOf(target), to find if a procedure exists. It's a little trickier than that. You say that the parser just skips over it, if a specific procedure is not found. But, to determine if a proc is not found, you need to check not only current this-procedure, but the super procedures too. Basically, you need to test something like this:

```
def var h as handle.
run proc_handler.p persistent set h. /* this contains the procedures for sax parsing */
this-procedure:add-super-procedure(h). /* add it as a super-procedure. */

/* do the parsing */
```

The proc handlers might be resolved from the super-procedures too.
In any case, what you need to do is add a ControlFlowOps.reachableInternalProcedure, which will use the WorkArea.inHandleResolvers like this:

```
   /**
    * Check if an internal procedure or function is reachable and valid. This searches the
    * specified handle and all the super-procedures.
    *
    * @param    h
    *           A valid procedure handle.
    * @param    name
    *           The procedure or function name.
    * @param    function
    *           Flag indicating if the search is for a function or for an internal procedure.
    * @param    modes
    *           The parameter modes.
    * @param    args
    *           The passed arguments.
    *
    * @return   null if the internal entry can't be found; false if
    *           found and arguments are invalid; true otherwise.
    */
   public static Boolean reachableInternalEntry(handle    h,
                                                String    name,
                                                boolean   function,
                                                String    modes,
                                                Object... args)
   {
      WorkArea wa = work.get();
      InternalEntryCaller caller = null;
      for (Resolver r : wa.inHandleResolvers)
      {
         caller = r.resolve(h, name, function, false, null);

         if (caller != null)
         {
            break;
         }
      }

      if (caller == null)
      {
         // was not found
         return null;
      }
```

```
      boolean valid = false;
      if (caller != null)
      {
         ArgValidationErrors argsValid = caller.valid(function, args);
         valid = (argsValid == ArgValidationErrors.NO_ERROR);

         if (valid)
         {
            // validate the parameter modes
            String definedModes = caller.getParameterModes(function);
            valid = modes.equals(definedModes);
         }
      }

      return valid;
   }
```

If this returns Boolean.TRUE, you can call ControlFlowOps.invokeIn.

- you are correct that a special message needs to be shown if parameters are invalid. But, do not rely on your own parameter validation, because that's already implemented in ControlFlowOps.

Post the testcases you've used to determine proc handler behavior here; I want to take a look at them.

Finally, you still have lots of out-of-standard code formatting problems; please try to adjust your coding to standards, as you write it.

#60 - 11/17/2013 06:20 PM - Evgeny Kiselev

Constantin Asofiei wrote:

> 2) ProcDefaultHandler is specify in update, but I've found that java SAX API is not correct. Here like SAX-WRITER we need to use StAX api. Because either if in 4gl we call parse method, 4gl parse all document via sax-parse-next and sax-parse-first so it's like stream API. that's why changing input source is possible. SAX API doesn't give opportunity to parse tag by tag. So in ProcDefaultHandler need to review end of class.

I don't understand what you mean here. Please post an example.

```
function testParse returns int (err-num as int):
   def var saxReader as handle.
   def var hHandle as handle.

   create sax-reader saxReader.
   saxReader:handler = this-procedure.

   def var isOK as logical.
```

```
   def var lc1 as longchar no-undo.
   def var c as character.

   lc1 = "<?xml version='1.0' ?><Phonelist><Entry ContactName='Jane Jones'> 555 555-5555 </Entry><Entry Contac
tName='John Smith'> 555 555-1111 </Entry></Phonelist>".

   saxReader:set-input-source("file", lc1).

   saxReader:sax-parse().
   /*message error-status:error.
   message error-status:num-messages.
   */
end.
testParse(-1).

procedure Characters:
   define input parameter charData AS longchar no-undo.
   define input parameter numChars AS INTEGER no-undo.

   message string(charData).
end procedure.

procedure STARTElement:
   define input parameter namespaceURI as character no-undo.
   define input parameter localName as character no-undo.
   define input parameter qname as character no-undo.
   define input parameter hAttributes as handle no-undo.

   message localName.
end procedure.

procedure Warning:
   define input parameter errMessage as character.
   message errMessage.
end procedure.

procedure FatalError:
   define input parameter errMessage as character.
   message errMessage.
end procedure.

procedure Error:
   define input parameter errMessage as character.
   message errMessage.
end procedure.
```

Because of wrong input-source it will not working, but exception will be not in method sax-parse, but in SAX-PARSE-FIRST. So calls of method sax-parse looks like:

```
saxReader:sax-parse-first() no-error.
repeat while saxReader:parse-status = sax-running:
  saxReader:sax-parse-next() no-error.
end.
```

So Idea is parse tag by tag. Java SAX API cannot parse tag by tag. that's why we need to use here StAX api.

Testcase for mismatched number of parameters passed to procedure:

```
function testParse returns int (err-num as int):
   def var saxReader as handle.
   def var hHandle as handle.

   create sax-reader saxReader.
   saxReader:handler = this-procedure.

   def var isOK as logical.
   def var lc1 as longchar no-undo.
   def var c as character.

   lc1 = "<?xml version='1.0' ?><Phonelist><Entry ContactName='Jane Jones'> 555 555-5555 </Entry><Entry Contac
```

```
tName='John Smith'> 555 555-1111 </Entry></Phonelist>".

   saxReader:set-input-source("longchar", lc1).

   saxReader:sax-parse().
   /*message error-status:error.
   message error-status:num-messages.
   */
end.
testParse(-1).

procedure STARTElement:
   define input parameter namespaceURI as integer no-undo.
   define input parameter localName as character no-undo.
   define input parameter qname as character no-undo.
   define input parameter hAttributes as handle no-undo.

   message localName.
end procedure.
```

#61 - 11/18/2013 02:52 AM - Constantin Asofiei

Evgeny Kiselev wrote:

> Constantin Asofiei wrote:
> So Idea is parse tag by tag. Java SAX API cannot parse tag by tag. that's why we need to use here StAX api.

OK, I understand now. What StAX implementation will you be using?

> Testcase for mismatched number of parameters passed to procedure:

I was correct, the search is done in super-procedures too. So, you can have two programs:

1. sax-handler.p, which can contain only callback implementations:

```
procedure Characters:
   define input parameter charData AS longchar no-undo.
   define input parameter numChars AS INTEGER no-undo.

   message string(charData).
end procedure.

procedure STARTElement:
   define input parameter namespaceURI as character no-undo.
   define input parameter localName as character no-undo.
   define input parameter qname as character no-undo.
   define input parameter hAttributes as handle no-undo.

   message localName.
end procedure.
```

```
procedure Warning:
   define input parameter errMessage as character.
   message errMessage.
end procedure.

procedure FatalError:
   define input parameter errMessage as character.
   message errMessage.
end procedure.

procedure Error:
   define input parameter errMessage as character.
   message errMessage.
end procedure.
```

2. test-sax.p, which adds sax-handler.p to its super-procedures list:

```
function testParse returns int (err-num as int):
   def var saxReader as handle.
   def var hHandle as handle.

   create sax-reader saxReader.
   saxReader:handler = this-procedure.

   def var isOK as logical.
   def var lc1 as longchar no-undo.
   def var c as character.

   lc1 = "<?xml version='1.0' ?><Phonelist><Entry ContactName='Jane Jones'> 555 555-5555 </Entry><Entry Co
ntactName='John Smith'> 555 555-1111 </Entry></Phonelist>".

   saxReader:set-input-source("longchar", lc1).

   saxReader:sax-parse().
   /*message error-status:error.
   message error-status:num-messages.
   */
end.
def var h as handle.
run sax-handler.p persistent set h.
this-procedure:add-super-procedure(h).
testParse(-1).
```

#62 - 11/18/2013 06:27 PM - Evgeny Kiselev

> OK, I understand now. What StAX implementation will you be using?

I'm using standard implementation in JDK.
http://docs.oracle.com/javase/tutorial/jaxp/stax/index.html

There is woodstox(standalone) implementation, but it's not necessary here.

#63 - 11/22/2013 05:12 AM - Evgeny Kiselev

Unfortunately we need to add woodstox StAX api. Because JDK StAX api is not support any validation (DTD and schemas). woodstox api has Apache license.
http://woodstox.codehaus.org/Download
What is the procedure to add new jars ? I've need to modify build script and place jars in appropriate place ?
Also we will not need any new development because woodstox StAX api works on the same interfaces and same base classes like JDK StAX api.

#64 - 11/22/2013 05:22 AM - Constantin Asofiei

Evgeny Kiselev wrote:

> Unfortunately we need to add woodstox StAX api. Because JDK StAX api is not support any validation (DTD and schemas). woodstox api has Apache license.

I guess you have tests showing that 4GL validates the XML against a DTD or a schema (when provided), right?

> http://woodstox.codehaus.org/Download
> What is the procedure to add new jars ? I've need to modify build script and place jars in appropriate place ?

You need to add the new jar to the manifest files in the p2j/manifest folder.

Greg: please advise which license to use (assuming at least one of them is OK for us), as woodstox comes in two flavours: LGPL 2.1 and Apache License 2.0.

#65 - 11/22/2013 06:17 AM - Evgeny Kiselev

Constantin Asofiei wrote:

> Evgeny Kiselev wrote:
>
> > Unfortunately we need to add woodstox StAX api. Because JDK StAX api is not support any validation (DTD and schemas). woodstox api has Apache license.
>
> I guess you have tests showing that 4GL validates the XML against a DTD or a schema (when provided), right?

Yes, 4GL validates the XML against a DTD or a schema (when provided).

#66 - 11/22/2013 09:42 AM - Greg Shah

If there is no difference in the versions, use the one that is Apache licensed.

#67 - 11/26/2013 11:39 AM - Greg Shah

How much more time do you estimate that you will need to complete this work?

#68 - 11/26/2013 06:42 PM - Evgeny Kiselev

Greg Shah wrote:

> How much more time do you estimate that you will need to complete this work?

I've finishing with xml validation now, and in my todo list next issues:
1) test callbacks error handle while error produced from callback/source or parser (6-10 hours)
2) finishing with sax-writer (6 hours)
3) and fix issues after review (? hours)

#69 - 11/29/2013 08:22 PM - Evgeny Kiselev

- File evk_upd20131129a.zip added

new drop for review:
1) Finishing with callback handler wrapper

2) added DTD and schema validation in SaxReader
3) added error handler (call callback's error procedure if some xml/validation errors)
4) fixed most of the code style problem
5) added javaDoc in most places

#70 - 11/30/2013 10:23 AM - Constantin Asofiei

Review for 1129a.zip; I like how you handled the callbacks in XMLCallbackWrapper and how SaxReaderImpl code looks like. At this point we are almost done with the reading, but the writing needs some more work/testing.
Misc stuff:

- add the StAX libraries to the manifest/p2jrt.mf file too.
- lots of files are missing history entries
- javadoc of ControlFlowOps.reachableInternalEntry is incosistent (see the return part)
- ADM-DATA and UNIQUE-ID I see are not implemented; please add support for them too. For UNIQUE-ID, you can use UniqueIdGenerator. Both can be handled by the SaxEntityImpl class, as this is the base class for SaxAttributes/Reader/Writer classes.
- when a c'tor/method doesn't have a body, please add a // no-op comment on a single line (so the reader knows the body is empty on purpose).

SaxEntity.java

- add an empty line after the Source enum.

SaxReader.java

- ReadStatus enum: leave an empty line between field feinitions.

SaxWriter.java

- WriteStatus enum: leave an empty line between field feinitions.

SaxAttributesImpl.java

- import statements should always import entire package, unless we need to disambiguate a class name
- updateAttribute has a TODO: check boundary

SaxEntityImpl.java

- I see you still use class-level constants for the legacy 4GL error numbers; from these, I see that only ERR_10511 and ERR_10512 are used more than once. So:
  - remove all these constants and inline them
  - for ERR_10511 and ERR_10512 cases, extract XMLCallbackWrapper$MismatchedParametersException in its own file and create another subclass (or maybe two), so that you have an unique exception for the 10511 (mismatched number of parameters) and 10512 (mismatched parameter types).

SaxReaderImpl.java

- please check how the imports are ordered
- is source field still needed, as it is never read, just assigned.
- improve the javadoc for validate and suppress fields
- why not keep the setter/getters for a certain attribute together? At least get/setSuppressNamespaceProcessing are split throught the class.
- please improve the javadoc for saxParse
- saxParseFirst - the No input file specified error is shown regardless of input source?
- setInputSource - for FILE mode, can't the source be a character value? Because you allow only a java string value.
- setInputSource - if this fails, make sure to invalidate all prior state (if 4GL does so); I refer to the readStatus/callbackWrapper and other fields. More, what happens if setInputSource is called while parsing is active?
- setInputSource - for STREAM-HANDLE and STREAM modes, log a "not supported" warning and initialize the inputStream so that parsing will not be possible (maybe a no-op InputStream implementation). Idea is, make sure inputStream is initialized properly for all possible modes.
- LongCharInputStream, FileInputStreamWrapper and MemptrInputStream classes:
  - should be at the end of the file
  - their fields definitions must be separated with an empty line
  - please document what happens if the source gets altered during parsing; at least for file mode, I think the changes are seen. Not sure about the longchar and memptr modes.
  - some methods are missing javadocs

SaxWriterImpl.java

- check the imports order/necessity (XmlHelper is not needed)
- sourceObject field is missing javadoc
- source and sourceObject I think are better named destination and destinationObject (because during writing we don't have a source, but a destination).
- default c'tor is missing javadoc.
- you have an unneeded new line in setOutputDestination; more, log a warning if STREAM or STREAM-HANDLE destinations are used and do

not allow writing.
- setOutputDestination - if this fails, make sure to invalidate all prior state (if 4GL does so); more, what happens if this is called while writing is active?
- are you sure the XML is sent all-at-once? Currently, I see that you write everything to a buffer and flush it to the destination only when endDocument is called. I think it might be possible for 4GL to send the XML document to the other side in pieces. Please post a test here so I can double check how this works. Better, commit all your tests to a uast/sax folder in the testcases project.
- throws declaration for toString(Node) is not formatted properly

XMLCallbackWrapper.java

- you need an empty line between the header and the package statement.
- double check the imports - some are not needed
- javadoc for isMemptrInCharacterCallback and isMemptrInIgnorableWhitespaceCallback should be improved
- good catch with checking which version of character/ignorableWhitespace to use
- invokeProcedure - we should log any exception encountered during the callback call, not ignore it.
- validateCallback is missing javadoc
- MismatchedParametersException needs empty lines between fields and the extends clause is formatted wrong.

#71 - 12/08/2013 09:01 PM - Evgeny Kiselev

- File evk_upd20131208a.zip added

> - the No input file specified error is shown regardless of input source?

Yes, it's not depends from source.

> setInputSource - if this fails, make sure to invalidate all prior state (if 4GL does so); I refer to the readStatus/callbackWrapper and other fields. More, what happens if setInputSource is called while parsing is active?

It's more complicated then I thought. For example for sax-writer it's just reset current buffer (everything that was written) and continue without any exceptions.
For sax-reader it doesn't matter at all. I.e. you can change input source but, sax parser will continue parse old source.
I need a little bit more time to retest it.

> ADM-DATA and UNIQUE-ID I see are not implemented; please add support for them too. For UNIQUE-ID, you can use UniqueIdGenerator. Both can be handled by the SaxEntityImpl class, as this is the base class for SaxAttributes/Reader/Writer classes.

I've implemented ADM-DATA and UNIQUE-ID, but I'm not sure about ADM-DATA. I haven't find any special behaviour of this attribute.
UNIQUE-ID is share for all SAX objects it 4gl.

#72 - 12/09/2013 03:54 AM - Constantin Asofiei

I like the part in SAX writer, we are getting close to finish this.

> For example for sax-writer it's just reset current buffer (everything that was written) and continue without any exceptions.

This is strange; lets say you are changing output destination from file a.xml to file b.xml in the middle of writing. You are saying that the resulted files may have an invalid structure?

> but I'm not sure about ADM-DATA. I haven't find any special behaviour of this attribute.

There is no special behaviour, this acts like PRIVATE-DATA - it just allows the user to save some additional info at the resource.

0. misc
- you need to merge with the latest revision
- in some files, you have an extra line at the end of the history entry.

1. MismatchedParametersException
- extract the two classes in their own files

2. SaxEntityImpl
- rename amdData field to admData
- setADMData must use something like this.admData.assign(value) - do not save references to external BDT objects, as they are mutable

3. SaxEntity
- history entry is misleading

4. SaxReaderImpl
- add an extra line after the LOG field
- getXMLEvent - throws clause is out of place
- please document what happens if the source gets altered during parsing; at least for file mode, I think the changes are seen. Not sure about the longchar and memptr modes.

5. SaxWriterImpl
- is the "com.sun.xml.internal.txw2.output.IndentingXMLStreamWriter" dependency needed? This looks like something vendor-specific, and I don't think is guaranteed to survive/exist between JVM versions.
- for comments like "Stream is not supporting" I think the correct version is "Stream is not supported"
- toString(Node) - the javadoc for throws is not formatted correct; should be:

```
    * @throws   TransformerFactoryConfigurationError
    *           If an error occurs writing the identifier.
    * @throws   TransformerException
    *           If an error occurs writing the identifier.
```

- about the wrapper classes; I see that only LongcharOutputStream writes to a buffer and flushes it; are you sure this is how 4GL behaves? If yes, did you determine the rules describing when the buffer is flushed, the buffer size, etc? If the data is really buffered, I suspect that in FILE mode, the underlying 4GL-like stream can take care of buffering/flushing. But what about the memptr mode - can this be buffered too?

6. XMLCallbackWrapper
- the static import for ControlFlowOps.ArgValidationErrors was OK, there was no reason to remove it.

#73 - 12/09/2013 07:27 PM - Evgeny Kiselev

Constantin Asofiei wrote:

> I like the part in SAX writer, we are getting close to finish this.
>
> > For example for sax-writer it's just reset current buffer (everything that was written) and continue without any exceptions.
>
> This is strange; lets say you are changing output destination from file a.xml to file b.xml in the middle of writing. You are saying that the resulted files may have an invalid structure?

I was not fully correct. I meant if we do something with output object (for example modify):

```
def var saxWriter as handle.
def var lc1 as longchar no-undo.
def var lc2 as longchar no-undo.
create sax-writer saxWriter.

saxWriter:set-output-destination("longchar", lc1).
saxWriter:start-document().
saxWriter:write-comment("asd").
message string(lc1).
lc1 = "".
saxWriter:write-comment("test2").
message string(lc1).
saxWriter:end-document().
```

will output:

```

<?xml version="1.0"?><!-- asd -->
<!-- test2 -->
```

Change output destination is impossible during writing. It will be threw exception of incorrect parser status.

> 4. SaxReaderImpl
> - please document what happens if the source gets altered during parsing; at least for file mode, I think the changes are seen. Not sure about the longchar and memptr modes.

will be check later

> 5. SaxWriterImpl
> - is the "com.sun.xml.internal.txw2.output.IndentingXMLStreamWriter" dependency needed? This looks like something vendor-specific, and I don't think is guaranteed to survive/exist between JVM versions.

Yes, it's not guaranteed in the future. I'll try to test jsr173 library, they have analogue functionality.

> - about the wrapper classes; I see that only LongcharOutputStream writes to a buffer and flushes it; are you sure this is how 4GL behaves? If yes, did you determine the rules describing when the buffer is flushed, the buffer size, etc? If the data is really buffered, I suspect that in FILE mode, the underlying 4GL-like stream can take care of buffering/flushing. But what about the memptr mode - can this be buffered too?

In 4gl it works in real time (at least after each element or tag everything is flushed to output source). For longchar I didn't find appropriate API to write byte(or char) by byte, that's why i wrote this constructions. Is it possible to extend longchar API ? or How I can write symbol(byte) into longchar ?
For memptr: I couldn't find particular moment then data is flushed to output object, but at least after every call of writing something, everything is flushed to output object.

For saxReader object it is possible to change input source in any time, but reader will continue to use old one if reading is already started.

#74 - 12/10/2013 06:56 AM - Constantin Asofiei

Evgeny Kiselev wrote:

> This is strange; lets say you are changing output destination from file a.xml to file b.xml in the middle of writing. You are saying that the resulted files may have an invalid structure?

I was not fully correct. I meant if we do something with output object (for example modify):

Thanks, now it makes sense. But what happens if the longchar var is set to unknown? I think the longchar var will remain unknown in the end... thus, the flush() in LongcharOutputStream should look like:

```
         String buf = buffer.getBuffer().toString();
         buffer.getBuffer().setLength(0);
         lchar.assign(TextOps.concat(lchar, buf));
```

This will do 4GL-style concatenation and will take care of unknown values properly.

> Change output destination is impossible during writing. It will be threw exception of incorrect parser status.

What happens if you call SET-INPUT-SOURCE after parsing was started?

> In 4gl it works in real time (at least after each element or tag everything is flushed to output source). For longchar I didn't find appropriate API to write byte(or char) by byte, that's why i wrote this constructions. Is it possible to extend longchar API ? or How I can write symbol(byte) into longchar ?

I understand, using a buffer is OK in this case.

> For saxReader object it is possible to change input source in any time, but reader will continue to use old one if reading is already started.

You mean to alter the content of the i.e. longchar var, right? If this is the case, what happens if the file or the memptr data gets changed/deleted AFTER the parsing was started (and not yet finished)?

#75 - 12/10/2013 06:46 PM - Evgeny Kiselev

Constantin Asofiei wrote:

> What happens if you call SET-INPUT-SOURCE after parsing was started?

> For saxReader object it is possible to change input source in any time, but reader will continue to use old one if reading is already started.

> You mean to alter the content of the i.e. longchar var, right? If this is the case, what happens if the file or the memptr data gets changed/deleted AFTER the parsing was started (and not yet finished)?

In 4gl it works very strange. I've tried on this example:

```
function testParse returns int (err-num as int):
   def var saxReader as handle.
   def var hHandle as handle.

   create sax-reader saxReader.
   saxReader:handler = this-procedure.

   def var isOK as logical.
   def var lc1 as longchar no-undo.
   def var lc2 as longchar no-undo.
   def var c as character.
   def var i as integer.

   lc1 = "<?xml version='1.0' ?><Phonelist><Entry ContactName='Jane Jones'> 555 555-5555
   </Entry><Entry ContactName='John Smith'> 555 555-1111 </Entry></Phonelist>".

   saxReader:set-input-source("longchar", lc1).
   saxReader:sax-parse-first().
   lc1 = ?.
   message "delete file at this moment" view-as alert-box.

   repeat while saxReader:parse-status = sax-running:
      saxReader:sax-parse-next().
      i = i + 1.
   end.

   message "lines: " i.

   message error-status:error.
   message error-status:num-messages.

end.
testParse(-1).
```

lc1 = ?. doesn't matter for parse at all. Parser will continue read old data in longchar. I though that it could be some buffering and tried to use big input data, but result was not changed. Same for "file" - if change file or choose different input source then parser will continue to use old input source.

Also I've tried the same example with files(I've deleted reading file after parsing is started and also tried to modify file). And result was the same too.

#76 - 12/10/2013 07:18 PM - Evgeny Kiselev

- File evk_upd20131210a.zip added

1) Fixed issues in note 72
2) Added stax-utils.jar (need to Indented output)
3) From note 74 - fixed problem with LongcharOutputStream if longchar become ? value during writing.

#77 - 12/11/2013 03:37 AM - Constantin Asofiei

A note about the update: the module in the header (for the new classes you've added) must be the file name, not the package name.

About the findings at note 75: I've checked with a large XML (~200MB) and, if the file is removed mid-parsing (after a sax-parse-next call), the XML is still parsed fully. Also, if the file is altered mid-parsing, the changes are not seen. Can't explain why they chose to implement it this way. A KB article: http://knowledgebase.progress.com/articles/Article/P11630 states that:

> When processing large XML file, consider using the SAX parser instead of the DOM interface. The SAX parser does not require loading the whole document into memory to process it.

If the longchar var or file is changed before the sax-parse-next call, the change will be seen.

Unfortunately, we will have to do it this way, too; but, please do some more testing to make sure we get this right.

#78 - 12/11/2013 08:09 AM - Greg Shah

What license is used for the stax-utils? Please put the URL to the project here.

#79 - 12/11/2013 04:33 PM - Evgeny Kiselev

Greg Shah wrote:

> What license is used for the stax-utils? Please put the URL to the project here.

https://java.net/projects/stax-utils/pages/Home
I've found license inside archive
https://java.net/projects/stax-utils/downloads/directory/Stable

#80 - 12/11/2013 07:37 PM - Evgeny Kiselev

Constantin Asofiei wrote:

> A note about the update: the module in the header (for the new classes you've added) must be the file name, not the package name.
>
> About the findings at note 75: I've checked with a large XML (~200MB) and, if the file is removed mid-parsing (after a sax-parse-next call), the XML is still parsed fully. Also, if the file is altered mid-parsing, the changes are not seen. Can't explain why they chose to implement it this way.
> A KB article: http://knowledgebase.progress.com/articles/Article/P11630 states that:
>
> > When processing large XML file, consider using the SAX parser instead of the DOM interface. The SAX parser does not require loading the whole document into memory to process it.
>
> If the longchar var or file is changed before the sax-parse-next call, the change will be seen.

longchar works OK in our implementation, because in first read we've copy content of longchar. This is works the same like in the 4gl.

I've try to find something with files (can't believe that they copy all file in memory). Maybe 4gl copy file in some temp directory, but I didn't find anything. Also I didn't find any big memory allocation with big xml files.
Maybe the key in something different like:

> A "file" under unix is an unnamed i-node (which contains data) and a zero+ entries in directories pointing to that i-node. Zero or more. Which means you can delete the name, but the i-node will still exist, and if open, one can write into it and read from it.

#81 - 12/12/2013 01:38 AM - Constantin Asofiei

> I've try to find something with files (can't believe that they copy all file in memory). Maybe 4gl copy file in some temp directory, but I didn't find anything. Also I didn't find any big memory allocation with big xml files.

Double check with memptr, by overwriting the in-memory XML data AFTER the sax-parse-next was called.

About files: it's pretty weird for them to go this low-level into accessing the file... have you confirmed the same behaviour on windev01?

#82 - 12/12/2013 06:30 AM - Greg Shah

The stax-utils license is this:

```
Copyright (c) 2004, Christian Niles, unit12.net
Copyright (c) 2004, Sun Microsystems, Inc.
Copyright (c) 2006, John Kristian
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

    * Redistributions of source code must retain the above copyright
      notice, this list of conditions and the following disclaimer.

    * Redistributions in binary form must reproduce the above
      copyright notice, this list of conditions and the following
      disclaimer in the documentation and/or other materials provided
      with the distribution.

    * Neither the name of the listed copyright holders nor the names
      of its contributors may be used to endorse or promote products
      derived from this software without specific prior written
      permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
```

This means that we must include this text in a file in the p2j/lib/ directory (put it in p2j/lib/stax-utils_license.txt).

#83 - 12/15/2013 07:47 PM - Evgeny Kiselev

- File evk_upd20131215a.zip added

1) added license
2) rollback constants in SaxReader and SaxWriter file (because this constants specify in literals.rules). I can change rules to work with new status Enumes.
3) memptr work different. Every changes in memptr always seen in real-time. So I've fixed in SaxReader and SaxWriter memptr reading/writer process.
4) SaxWriter: Memeptr and longchar will cleared before first write in this objects.

#84 - 12/16/2013 03:25 AM - Constantin Asofiei

Review for 1215a.zip:
SaxReader/SaxWriter
- add back the public final static modifiers to the sax constants. I know the interface fields are public final static by default, but it's easier to read them if they are explicitly set. Do not change the conversion rules for these constants now.
SaxReaderImpl
- MemptrInputStream.read - is it possible to save the mptr.length() value, to avoid extra trips into native calls? I guess this should be possible only if the memptr can't be resized during parsing (or at least the size change is not seen).
SaxWriterImpl
- line 222 - you have an incorrect javadoc: Gets the valustricte of the ...
- toString() - throw's javadoc is formatted incorrect
SaxWriter
- the alignment of the enum elements is out of sync

Please add javadoc to the custom input and output stream wrappers, in terms of implemented 4GL behaviour. Beside this, please summarize what else needs to be done with this task.

#85 - 12/16/2013 08:09 AM - Constantin Asofiei

Evgeny Kiselev wrote:

> I've try to find something with files (can't believe that they copy all file in memory). Maybe 4gl copy file in some temp directory, but I didn't find anything. Also I didn't find any big memory allocation with big xml files.
> Maybe the key in something different like:
>
> > A "file" under unix is an unnamed i-node (which contains data) and a zero+ entries in directories pointing to that i-node. Zero or more. Which means you can delete the name, but the i-node will still exist, and if open, one can write into it and read from it.

I think you are correct. Even if a file gets deleted, the i-node is not deleted until the process using it is terminated.

#86 - 12/22/2013 08:07 PM - Evgeny Kiselev

- File evk_upd20131222a.zip added

Constantin Asofiei wrote:

> Review for 1215a.zip:
> - MemptrInputStream.read - is it possible to save the mptr.length() value, to avoid extra trips into native calls? I guess this should be possible only if the memptr can't be resized during parsing (or at least the size change is not seen).

In MemptrInputStream.read memptr length is constant and never change(either other changes is vis

> Please add javadoc to the custom input and output stream wrappers, in terms of implemented 4GL behaviour. Beside this, please summarize what else needs to be done with this task.

Most of cases I've tested and I think everything is already done. Only need final review and regression testing.

#87 - 12/23/2013 03:07 AM - Constantin Asofiei

Evgeny, the logic looks good, I don't have any other comments beside some formatting/javadoc issues and some left-behind TODOs. See bellow for he review of 1222a.zip:

- make sure all imports use wildcard (if not otherwise required)
- you have some javadoc errors in handle.java and SaxReaderImpl - please fix them
- SaxAttributesImpl
  - you have an extra empty line at 1154
- SaxEntityImpl
  - getADMData - an unneeded empty line
  - you have some TODOs left in insertAttribute and removeAttribute: these APIs always return true, and their body is a no-op. Is this correct?
  - some throws statements are not on their own line: 528, 934
- SaxWriterImpl
  - serializeNodeToString - the throws javadoc is formatted incorrect, put the comment on a new line, not after the exception
- XMLCallbackWrapper
  - the javadoc for the fields is formatted incorrect. Multi-line comment needs to be like this:

```
/**
 * long-line 1
 * long-line 2
 * long-line 3
 */
private final handle field;
```

- @throws XMLStreamException is missing javadoc in most (all?) methods
- when you add a see javadoc, make sure there are empty lines before it and after it (as needed). same for throws javadoc - add an empty line before it.
- after endElemen, you have two empty lines
- invokeProcedure, fatalError have no javadoc for return
- you have an unneeded empty line at 451.

After you fix these, you can go ahead and get it regression tested.

#88 - 12/24/2013 07:33 PM - Evgeny Kiselev

- File 01_evk_upd20131223a.zip added

This update has passed regression.

But I have found that I forgot implementation of NONAMESPACE-SCHEMA-LOCATION attribute.
I'm working on it now and will upload update and run regression tests again.

#89 - 12/31/2013 08:02 PM - Evgeny Kiselev

- File 01_evk_upd20131231a.zip added

1) added DTD validation with specific schema-paths
2) added XSD schema validation
3) added callback resolveEntity procedure support

Regression tests has been passed.

#90 - 01/03/2014 04:15 AM - Constantin Asofiei

Review for 1231a.zip:

1. what are licences for the new jars?

```
xsdlib-2010.1.jar
woodstox-msv-rng-datatype-20020414.jar
msv-core-2010.2.jar
```

2. is there some reason you are moving from wildcard import to explicit class import in SaxAttributesImpl?
3. SaxReaderImpl.saxParseFirst line 444 is formatted wrong; should be like this:

```
                    if ("http".equals(protocol) || "https".equals(protocol) ||
                        "ftp".equals(protocol) || "ftps".equals(protocol))
```

   Also, are you sure the match is done case sensitive (same for addValidation:862)?
4. SaxReaderImpl: getNonamespaceSchemaLocation and setNonamespaceSchemaLocation are missing javadoc.
5. SaxWriterImpl: line 2530 is garbage.
6. you have renamed XmlSchema.getSchemaPath to XmlSchema.getSchemaPaths. You need to be careful when renaming APIs which are emitted by conversion rules. In this case, there should have been some changes in convert/methods_attributes.rules, so that the SCHEMA-PATH attribute is converted properly. How come didn't you caught this in your testcases? Anyway, I think is best to leave XmlSchema.getSchemaPath as is and back out this rename.
7. the javadoc errors are still there.

#91 - 01/04/2014 12:45 PM - Evgeny Kiselev

Constantin Asofiei wrote:

> Review for 1231a.zip:
>
> 1. what are licences for the new jars?

under BSD license. It was MSV project from SUN.
https://java.net/projects/msv/sources/svn/show/trunk/msv?rev=1827

1. is there some reason you are moving from wildcard import to explicit class import in SaxAttributesImpl?
2. you have renamed XmlSchema.getSchemaPath to XmlSchema.getSchemaPaths. You need to be careful when renaming APIs which are emitted by conversion rules. In this case, there should have been some changes in convert/methods_attributes.rules, so that the SCHEMA-PATH attribute is converted properly. How come didn't you caught this in your testcases? Anyway, I think is best to leave XmlSchema.getSchemaPath as is and back out this rename.
3. the javadoc errors are still there.

It was mistake in my IDE, I've made refactoring and then rollback changes, but didn't look into careful of result.

#92 - 01/04/2014 02:56 PM - Evgeny Kiselev

- File evk_upd20140104a.zip added

fixed issues from note 90

#93 - 01/06/2014 07:42 AM - Constantin Asofiei

Review for 0104a.zip:

- SaxAttributesImpl.removeAttribute (both versions): javadoc for the parameters is not on distinct lines
- the javadoc errors in SaxReaderImpl.java are still there:

  [javadoc] /working/workspace/p2j4sync/src/com/goldencode/p2j/xml/SaxReaderImpl.java:266: warning - Tag @link: reference not found: ReadStatus#SAX_UNINITIALIZED
  [javadoc] /working/workspace/p2j4sync/src/com/goldencode/p2j/xml/SaxReaderImpl.java:266: warning - Tag @link: reference not found: ReadStatus#SAX_COMPLETE
  [javadoc] /working/workspace/p2j4sync/src/com/goldencode/p2j/xml/SaxReaderImpl.java:266: warning - Tag @link: reference not found: ReadStatus#SAX_PARSER_ERROR
  [javadoc] /working/workspace/p2j4sync/src/com/goldencode/p2j/xml/SaxReaderImpl.java:266: warning - Tag @link: reference not found: ReadStatus#SAX_RUNNING

Otherwise it looks good.

#94 - 01/06/2014 07:41 PM - Evgeny Kiselev

- File 01_evk_upd20140106a.zip added

1) merged with latest revision(removed SAX_ATTRIBUTES and SAX_READER from IdKind enum, because all SAX objects are shared common id sequence)
2) fixed javaDoc issues

#95 - 01/07/2014 03:39 AM - Constantin Asofiei

Did this pass testing? If yes, then this is next on queue to be committed, unless Greg has any objections.

BTW, I can't find the tests for the DOM/XML and SAX tasks on the testcases/ repository; please add them there.

#96 - 01/07/2014 08:24 AM - Greg Shah

I have no objections. We need to get this committed as soon as it passes regression testing.

> BTW, I can't find the tests for the DOM/XML and SAX tasks on the testcases/ repository; please add them there.

Please place them in testcases/uast/xml/dom/ and testcases/uast/xml/sax/ directories.

#97 - 01/07/2014 06:02 PM - Evgeny Kiselev

Passed the regression testing. Committed at bzr revision 10428.

#98 - 01/07/2014 06:46 PM - Evgeny Kiselev

Greg Shah wrote:

> Please place them in testcases/uast/xml/dom/ and testcases/uast/xml/sax/ directories.

DOM testcases committed at bzr revision 1079.
SAX testcases will be committed later. I need clean up some testcases.

#99 - 01/08/2014 08:36 AM - Constantin Asofiei

Evgeny, we missed something: deletion of the SAX and DOM/XML objects. How I've noticed this: the XCommonImpl.resourceDelete and XEntityImpl.valid APIs are not overridden by either the SAX or DOM/XML related classes. By default, it returns false or true (for valid), thus these resources will not be deleted by DELETE OBJECT statements and they always remain valid.

In the Web Services documentation (table on page 9-27) it states that DELETE OBJECT xNodeRefHandle. will not delete the referred XML DOM object, but a DELETE OBJECT xDocumentHandle. will delete all XML DOM Objects (so what happens with any X-NODEREF resources? do they get deleted? or only the underlying DOM node gets removed/invalidated?).

So, you need to check the DELETE OBJECT statement with all the SAX and DOM/XML resources. Implementation may be as simple as setting a deleted flag at the resource, so a valid call will return it (you will need to override the XEntityImpl.valid and XCommonImpl.resourceDelete APIs). Post your findings here.

#100 - 01/08/2014 07:50 PM - Evgeny Kiselev

After deleting X-DOCUMENT all nodes become uninitialized.

```
   def var hDoc as handle.
   def var hDoc2 as handle.
   def var hRoot as handle.
   def var hChild as handle.
   def var hTemp as handle.
   def var res as logical init false.
   def var c as character init false.
   create x-document hDoc.
   create x-document hDoc2.
   create x-noderef hChild.
   create x-noderef hTemp.
   create x-noderef hRoot.

   hDoc:create-node(hRoot,"root","element").
   hDoc:append-child(hRoot).
   hDoc:create-node(hChild,"child","element").
   hRoot:append-child(hChild).

   if not valid-handle(hDoc)
      then message "Unexpected error hDoc: is invalid".
   hRoot:get-parent(hTemp).
   if not hTemp:subtype <> "DOCUMENT" then "Unexpected error: document should in hTemp".

   delete object hDoc.
   if valid-handle(hDoc)
      then message "Unexpected error: hDoc should be invalid".
   if not valid-handle(hRoot)
         then message "Unexpected error: hRoot is invalid".

   /* all nodes become uninitialized and unknown, (removed from hDoc) */
   hRoot:name no-error.
   if error-status:error or error-status:num-messages <> 1 or not error-status:get-number(1) eq 9102
                     then message "Unexpected error get-parent: " error-status:get-number(1) " " error-status:
get-message(1).
```

After deleting X-NODEREF only a handle is deleted. Node is still available in the document and all children for this node are available too.

```
   def var hDoc as handle.
   def var hDoc2 as handle.
   def var hRoot as handle.
   def var hChild as handle.
   def var hTemp as handle.
   def var hTemp2 as handle.
   def var res as logical init false.
   def var c as character init false.
   create x-document hDoc.
   create x-document hDoc2.
   create x-noderef hChild.
   create x-noderef hTemp.
   create x-noderef hTemp2.
   create x-noderef hRoot.

   hDoc:create-node(hRoot,"root","element").
   hDoc:create-node(hChild,"child","element").
   hDoc:append-child(hRoot).
   hRoot:append-child(hChild).

   delete object hRoot.
   if valid-handle(hRoot)
      then message "Unexpected error: hRoot should be invalid".
   hDoc:get-child(hTemp, 1).
   if not hTemp:name <> "root" then "Unexpected error: root element should be exist in document".
```

```
hTemp:get-child(hTemp2, 1).
if not hTemp:name <> hTemp2:name then "Unexpected error: children names should be equal".
```

SAX-ATTRIBUTES delete only resource. it doesn't appear to sax-reader or sax-writer.
In SAX-READER and SAX-WRITER need to close all input/output streams.

Plan:
1) in XDocumentImpl override resourceDelete() and implement right deletion.
2) Should XEntityImpl.valid return false after deletion ?
3) in XEntityImpl override resourceDelete() and make xnode = null.
4) in SaxEntityImpl SaxWriterImpl and override resourceDelete() and implement clean up procedure.

#101 - 01/09/2014 03:02 AM - Constantin Asofiei

Evgeny Kiselev wrote:

> After deleting X-DOCUMENT all nodes become uninitialized.

You've checked only the root X-NODEREF, but you need to check all created X-NODEREF resources which are linked with a deleted X-DOCUMENT, to make sure valid-handle and attribute access behaves properly.

> 1) in XDocumentImpl override resourceDelete() and implement right deletion.

Note that for SOAP Header resource, I found that the same DOM XML node may be referred by multiple SOAP Header Entry resources. And,when deleting the SOAP Header, all DOM XML nodes are deleted (thus all SOAP Header Entry resources become uninitialized). As you noted, After deleting X-DOCUMENT all nodes become uninitialized.; so, you need a way to invalidate all X-NODEREF resources which refer DOM XML nodes in the deleted X-DOCUMENT; you may be able to add some user-data at the owner-document (i.e. to mark the full DOM Document as deleted), and check that user-data, when a XNoderefImpl accesses its DOM XML node.

> 2) Should XEntityImpl.valid return false after deletion ?

You need to make sure that after a DELETE OBJECT call, the target resource (DOM XML and SAX) is reported correctly by VALID-HANDLE function. So, after deletion is successful, you need valid to return false.

#102 - 01/12/2014 08:59 PM - Evgeny Kiselev

- File evk_upd20140112a.zip added

added correct DELETE operation for DOM/SAX.

#103 - 01/13/2014 02:59 AM - Constantin Asofiei

Review for 0112a.zip:

- all files are missing history entries. When you add history entries, don't forget to update the copyright years too (i.e. 2013-2014)
- XDocumentImpl - are you sure that if the X-DOCUMENT resource has no DOM document attached, resourceDelete just returns false? This will not allow an uninitialized X-DOCUMENT to be deleted...
- XEntityImpl
  - isInitialized: on line 822, just return true, as you no longer need to call getNode() again.
  - please add a package private constant XEntityImpl.IS_INVALID_NODE, and use it in isInitialized and XDocument.resourceDelete (instead of hard-coding the "IS_INVALID_NODE" string).

#104 - 01/14/2014 08:40 PM - Evgeny Kiselev

- File evk_upd20140113a.zip added

Constantin Asofiei wrote:

> Review for 0112a.zip:
>
> - all files are missing history entries. When you add history entries, don't forget to update the copyright years too (i.e. 2013-2014)

fixed

> - XDocumentImpl - are you sure that if the X-DOCUMENT resource has no DOM document attached, resourceDelete just returns false? This will not allow an uninitialized X-DOCUMENT to be deleted...

X-DOCUMENT is always initialized. Only during impossible exception in constructor it may becomes in uninitialized. Fixed this problem.

> - XEntityImpl
>   - isInitialized: on line 822, just return true, as you no longer need to call getNode() again.

fixed. Removed method call resourceDelete in isInitialized. Because it was incorrect. Need only make node uninitialized.

> - please add a package private constant XEntityImpl.IS_INVALID_NODE, and use it in isInitialized and XDocument.resourceDelete (instead of hard-coding the "IS_INVALID_NODE" string).

done.

#105 - 01/15/2014 01:53 AM - Constantin Asofiei

Review for 0113a.zip: the update is OK, but please make the XCommonImpl.isValid field protected instead of package-private (there is no need to expose it outside of the hierarchy).

If you are satisfied with your tests, go ahead and release it (after making the change above); ~~please~~ place the released update here too. There is no need for runtime/conversion testing as the changes are not reached by MAJIC code.

#106 - 01/15/2014 07:39 PM - Evgeny Kiselev

- File evk_upd20140115a.zip added

Committed to bzr revision 10438.

#107 - 01/16/2014 06:55 AM - Greg Shah

Please check in your SAX testcases.

#108 - 01/23/2014 08:33 PM - Evgeny Kiselev

SAX testcases committed at bzr revision 1088.

#109 - 01/23/2014 09:47 PM - Greg Shah

- Status changed from WIP to Closed

#110 - 11/16/2016 11:42 AM - Greg Shah

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

Files

sax_xml_test.p.20130211.zip	1.58 KB	02/11/2013	Eugenie Lyzenko
evl_upd20130114a.zip	85.5 KB	02/15/2013	Eugenie Lyzenko
sax_xml_test.p.20130214a.zip	2.6 KB	02/15/2013	Eugenie Lyzenko
evl_upd20130215a.zip	86.5 KB	02/15/2013	Eugenie Lyzenko
evl_upd20130215b.zip	87.5 KB	02/16/2013	Greg Shah
ca_upd20130319b.zip	10.5 KB	03/19/2013	Constantin Asofiei
ca_upd20130319c.zip	354 Bytes	03/19/2013	Constantin Asofiei
evl_upd20130320a.zip	61.2 KB	03/20/2013	Eugenie Lyzenko
evl_upd20130320b.zip	61.5 KB	03/20/2013	Eugenie Lyzenko
evk_upd20130922a.zip	138 KB	10/08/2013	Evgeny Kiselev
evk_upd20131020a.zip	30.4 KB	10/21/2013	Evgeny Kiselev
evk_upd20131030a.zip	46.6 KB	10/31/2013	Evgeny Kiselev
evk_upd20131105a.zip	50.1 KB	11/06/2013	Evgeny Kiselev
evk_upd20131115a.zip	52.2 KB	11/15/2013	Evgeny Kiselev
evk_upd20131129a.zip	693 KB	11/30/2013	Evgeny Kiselev
evk_upd20131208a.zip	691 KB	12/09/2013	Evgeny Kiselev
evk_upd20131210a.zip	814 KB	12/11/2013	Evgeny Kiselev
evk_upd20131215a.zip	818 KB	12/16/2013	Evgeny Kiselev
evk_upd20131222a.zip	818 KB	12/23/2013	Evgeny Kiselev
01_evk_upd20131223a.zip	818 KB	12/25/2013	Evgeny Kiselev
01_evk_upd20131231a.zip	1.58 MB	01/01/2014	Evgeny Kiselev
evk_upd20140104a.zip	1.57 MB	01/04/2014	Evgeny Kiselev
01_evk_upd20140106a.zip	1.57 MB	01/07/2014	Evgeny Kiselev
evk_upd20140112a.zip	29.6 KB	01/13/2014	Evgeny Kiselev
evk_upd20140113a.zip	29.9 KB	01/15/2014	Evgeny Kiselev