Base Language - Feature #4828

rework ErrorManager to use the TransactionHelper design on the server and a sync'd state design on the client

07/30/2020 08:58 AM - Greg Shah

Status:	Test	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:	Greg Shah	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:			
billable:	No	**vendor_id:**	GCD

Description

History

#1 - 07/30/2020 09:03 AM - Greg Shah

This task is for performance improvement.

Server-side usage of the EM can be expensive. For example, the isPendingError() is called often and requires context-local lookups. The TransactionHelper approach can eliminate some overhead on the server.

The client-side approach to the EM is even more expensive because of the round trip to the server for many calls. Constantin's profiling of a customer application screen showed that isHeader() and isIgnore() are called thousands of times. Moving to a sync'd state approach (adding state to the ServerState and applying the changes to the client side) can eliminate much of this usage.

#2 - 07/30/2020 09:03 AM - Greg Shah

- Status changed from New to WIP

- Assignee set to Greg Shah

#3 - 08/20/2020 12:08 PM - Greg Shah

- Status changed from WIP to Test

- % Done changed from 0 to 100

3821c revision 11460 greatly reduces the most common context-local lookups for ErrorManager on both the server and client sides. The new approach implements the ErrorManager.ErrorHelper inner class for use on the server-side only. This can be accessed directly or as part of the TransactionManager.TransactionHelper. By using this it avoids a context-local lookup to do common things like checking the silent or pending error flags. Some client side up calls were eliminated by allowing the state for the header and ignore flags to be managed locally on the client for the duration of the processing. At this time, it was not needed to use ServerState and the state synchronization approach to allow the upcalls to be removed. That is a future possible improvement.

#4 - 08/20/2020 04:24 PM - Eugenie Lyzenko

Greg,

The rev 11460 causes abend with ClassCastException regression with current customer application:

```
...
```

```
Caused by: java.lang.ClassCastException: com.sun.proxy.$Proxy6 cannot be cast to com.goldencode.p2j.util.Error
Manager$ServerDataAccess
        at com.goldencode.p2j.util.ErrorManager.getErrorHelper(ErrorManager.java:2556)
        at com.goldencode.p2j.util.TransactionManager$ContextContainer.obtain(TransactionManager.java:10469)
        at com.goldencode.p2j.util.TransactionManager.setNestedMode(TransactionManager.java:2756)
        at com.goldencode.p2j.util.Stream.putWorker(Stream.java:5634)
        at com.goldencode.p2j.util.StreamDaemon.putWorker(StreamDaemon.java:1128)
        at com.goldencode.p2j.util.LowLevelStreamMethodAccess.invoke(Unknown Source)
        at com.goldencode.p2j.util.MethodInvoker.invoke(MethodInvoker.java:156)
        at com.goldencode.p2j.net.Dispatcher.processInbound(Dispatcher.java:755)
        at com.goldencode.p2j.net.Conversation.block(Conversation.java:412)
        at com.goldencode.p2j.net.Conversation.waitMessage(Conversation.java:348)
        at com.goldencode.p2j.net.Queue.transactImpl(Queue.java:1201)
        at com.goldencode.p2j.net.Queue.transact(Queue.java:672)
        at com.goldencode.p2j.net.BaseSession.transact(BaseSession.java:271)
        at com.goldencode.p2j.net.HighLevelObject.transact(HighLevelObject.java:211)
        at com.goldencode.p2j.net.RemoteObject$RemoteAccess.invokeCore(RemoteObject.java:1473)
        at com.goldencode.p2j.net.InvocationStub.invoke(InvocationStub.java:145)
        at com.sun.proxy.$Proxy3.standardEntry(Unknown Source)
        at com.goldencode.p2j.main.ClientCore.start(ClientCore.java:380)
        at com.goldencode.p2j.main.ClientCore.start(ClientCore.java:167)
        at com.goldencode.p2j.main.ClientDriver.start(ClientDriver.java:250)
        at com.goldencode.p2j.main.CommonDriver.process(CommonDriver.java:444)
        at com.goldencode.p2j.main.ClientDriver.process(ClientDriver.java:144)
        at com.goldencode.p2j.main.ClientDriver.main(ClientDriver.java:313)
...
```

#5 - 08/20/2020 08:19 PM - Greg Shah

3821c rev 11464 should resolve the ClassCastException.