

Database - Feature #2148

Feature # 1656 (Closed): add Microsoft SQL Server support

enhance prototype to compile p2jpl.jar file with IKVMC into .NET assembly and load into SQL Server

05/02/2013 10:17 PM - Eric Faulhaber

Status:	Closed	Start date:	09/06/2013
Priority:	Normal	Due date:	09/27/2013
Assignee:	Ovidiu Maxiniuc	% Done:	100%
Category:		Estimated time:	100.00 hours
Target version:	Cleanup and Stabilization for Server Features		
billable:	No	vendor_id:	GCD
Description			
Related issues:			
Blocked by Database - Feature #2143: prototype IKVM solution for Java UDFs co...		Closed	08/26/2013 09/05/2013

History

#1 - 05/02/2013 10:29 PM - Eric Faulhaber

- Estimated time changed from 60.00 to 100.00

- Due date changed from 09/18/2013 to 09/27/2013

The scope of this issue is to take the implementation from prototype to a working solution which supports all of the UDFs we currently support under PL/Java. This will involve removing as many unnecessary classes from p2jpl.jar as is possible by unraveling dependencies that don't make sense in the target environment (e.g., TransactionManager, BlockManager, SecurityManager, etc.). The p2jpl.jar should compile fully into CIL and all UDFs should work.

The needs of the Java UDFs we use are not complicated or esoteric (e.g., math, string manipulation), so they should be well supported by IKVM. By removing the more complicated classes (which are not needed but are loaded by bogus dependencies), we should have good level of confidence in the resulting CIL. A nice side effect is that the p2jpl.jar will be simplified for PL/Java, too.

#2 - 03/17/2014 02:40 PM - Ovidiu Maxiniuc

- Status changed from New to WIP

#3 - 03/17/2014 04:30 PM - Eric Faulhaber

- Assignee changed from Eric Faulhaber to Ovidiu Maxiniuc

#4 - 04/11/2014 02:25 PM - Ovidiu Maxiniuc

- File om_upd20140314a.zip added

Using the attached om_upd20140314a.zip, almost all methods and functions have full support for MSSQL (the fully implemented Operators.cs and Functions.cs will defer the calls to our Operators.java and Functions.java respectively). However, except for some late additions (like plusMillis methods), I believe that these operators are used only in some specific configuration (I remember I played around with it, but at this moment I don't remember the exact location of the switch).

The p2jpl.jar is a little too oversized at this moment. I should investigate what should be removed and, more than that, some code could be rewritten because there are some restrictions, don't know if because of CLR or IKVM, but I was forced to install the p2j2clr as unsafe because of some constraints like this one:

Msg 6522, Level 16, State 2, Line 1

A .NET Framework error occurred during execution of user-defined routine or aggregate "toDate_t":
System.Security.HostProtectionException: Attempted to perform an operation that was forbidden by the CLR host.

The protected resources (only available with full trust) were: All
The demanded resources were: MayLeakOnAbort

```
System.Security.HostProtectionException:  
  at System.RuntimeMethodHandle.PerformSecurityCheck(Object obj, RuntimeMethodHandleInternal method, RuntimeType  
type parent, UInt32 invocationFlags)  
  at System.RuntimeMethodHandle.PerformSecurityCheck(Object obj, IRuntimeMethodInfo method, RuntimeType paren  
t, UInt32 invocationFlags)  
  at System.Reflection.RuntimeMethodInfo.Invoke(Object obj, BindingFlags invokeAttr, Binder binder, Object[]  
parameters, CultureInfo culture)  
  at System.Reflection.RuntimePropertyInfo.GetValue(Object obj, BindingFlags invokeAttr, Binder binder, Objec  
t[] index, CultureInfo culture)  
  at System.Reflection.RuntimePropertyInfo.GetValue(Object obj, Object[] index)  
  at IKVM.NativeCode.java.util.TimeZone.GetCurrentTimeZoneID()  
  at IKVM.NativeCode.java.util.TimeZone.getSystemTimeZoneID(String javaHome, String country)  
  at java.util.TimeZone.setDefaultZone()  
  at java.util.TimeZone.getDefaultRef()  
  at java.util.Date..ctor(Int32 year, Int32 month, Int32 date, Int32 hrs, Int32 min, Int32 sec)  
  at java.sql.Timestamp..ctor(Int32 year, Int32 month, Int32 date, Int32 hour, Int32 minute, Int32 second, In  
t32 nano)  
  at Common.toTimestamp(DateTime t)  
  at Functions.toDate_t(DateTime dt)
```

It's not clear to me where and why this leak may occur but I am thinking that there could be other issues like this out there waiting to be discovered. What's strange is that the stack points to a simple getter (GetCurrentTimeZoneID()) that should not normally allocate resources that could eventually leak.

Also there are lots of warnings in the ikvmc output log that annoy me.

#5 - 08/01/2014 09:50 AM - Eric Faulhaber

What is the status of the issues described in and the update attached to note 4? This effort seems to have merged with #2332. Has this update been superseded/obsoleted by the recent updates (0729b & c) in #2332?

Assuming those other updates review well, can this issue be closed?

#6 - 08/01/2014 10:20 AM - Ovidiu Maxiniuc

Eric Faulhaber wrote:

What is the status of the issues described in and the update attached to note 4? This effort seems to have merged with #2332. Has this update been superseded/obsoleted by the recent updates (0729b & c) in #2332?

Assuming those other updates review well, can this issue be closed?

Indeed, I merged this updated with the latest update from #2332.

Maybe we should keep this issue as a placeholder for cleanup changes that shortens the list of warnings issued when converting p2jpl.jar to pl2pl.dll ?

Most of them are 'broken' dependencies, here are some examples (1st encounter) of each of the three warning types:

```
warning IKVMC0105: Unable to compile class "com.goldencode.expr.ExpressionLexer"
    (missing class "antlr.CharScanner")
warning IKVMC0100: Class "com.goldencode.expr.ExtraAst" not found
warning IKVMC0111: Emitted java.lang.NoClassDefFoundError in "com.goldencode.expr.Compiler.parse()Lcom.goldenc
ode.expr.ExtraAst;"
    ("com.goldencode.expr.ExpressionLexer")
```

I guess some refactoring of the classes from p2jpl.jar could remove these but I cannot estimate the work volume for this.

#7 - 03/25/2016 04:55 PM - Eric Faulhaber

Ovidiu Maxiniuc wrote:

Maybe we should keep this issue as a placeholder for cleanup changes that shortens the list of warnings issued when converting p2jpl.jar to pl2pl.dll?

Is there any known functional impact to leaving these warnings? As you said, these all seem like broken dependencies. We don't need any of the classes noted in the example warnings you posted above for UDF support, but are these dependencies preventing anything we do need from compiling?

If the warnings are indicative of a real problem that is breaking some aspect of our UDF support, we need to fix them. If not, let's close this issue and open a separate task to track these, but they don't need to be fixed for us to consider the UDF support functionally complete.

#8 - 03/25/2016 05:29 PM - Ovidiu Maxiniuc

- File *p2jpl_conv.errlog* added

Firstly, just as an update: with IKVM.NET 8.1, the warning are a little changed, the IKVMC0105 & IKVMC0111 are gone, only IKVMC0100 are logged. However, the problem remains the same: queries will fail if one of the class need to be accessed (I believe the event is logged by SQL Server). On the other side, we normally don't access those classes at runtime. The Operators and Functions only do basic data manipulation. If the execution path reached the listed classes, I think it's a bug and will have to be corrected when found. Let's close this issue now.

I attached the listing from one of my latest conversion, with IKVM.NET 8.1, just for reference.

#9 - 03/28/2016 12:07 AM - Eric Faulhaber

- % Done changed from 0 to 100

- Status changed from WIP to Closed

Please open a separate issue to clean up the dependencies, so we at some point eliminate this noise from the build.

I agree that those classes should never be called by the basic data manipulation done in Operators and Functions. Closing this issue accordingly.

#10 - 11/16/2016 12:06 PM - Greg Shah

- Target version changed from Milestone 11 to Cleanup and Stablization for Server Features

Files

om_upd20140314a.zip	126 KB	04/11/2014	Ovidiu Maxiniuc
p2jpl_conv.errlog	15.7 KB	03/25/2016	Ovidiu Maxiniuc