

Base Language - Bug #2070

BlockManager Action disambiguation

02/28/2013 10:35 AM - Costin Savin

Status:	Closed	Start date:	02/28/2013
Priority:	Normal	Due date:	
Assignee:	Costin Savin	% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:	Conversion Support for Server Features	case_num:	
billable:	No		
vendor_id:	GCD		
Description			

History

#1 - 02/28/2013 10:40 AM - Costin Savin

When we have a progress file named action and it contains a block which will emit a reference to the Action enum of the BlockManager class the conversion result will be ambiguous and compile will fail.

ex:

```
def var i as integer.  
repeat i = 1 to 5 on error undo, leave:  
end
```

The result will be:

```
...  
OnPhrase[] onPhrase0 = new OnPhrase[]  
{  
    new OnPhrase(Condition.ERROR, Action.LEAVE, "loopLabel0")  
};  
...
```

#2 - 02/28/2013 10:43 AM - Costin Savin

Do you think it's better to replace Action.<enum_val> with BlockManager.Action.<enum_val> allways or check file name to see if it's equal to action and only replace for this case.

#3 - 02/28/2013 11:39 AM - Costin Savin

- File *cs_upd20130228a.zip* added

Added proposed update

#4 - 02/28/2013 11:47 AM - Greg Shah

This is fine for now. Ultimately, we will need to do this on a more global scale for all names. Please get it conversion tested.

#5 - 02/28/2013 02:12 PM - Constantin Asofiei

Costin, looks like we need another check: if there is a DMO class with the Action name, then qualify Action with BlockManager.Action. Is not enough to disambiguate current class.

#6 - 02/28/2013 02:35 PM - Costin Savin

Can the "classname" annotation of BUFFER_SCOPE be used safely to get that checked?

#7 - 02/28/2013 02:51 PM - Constantin Asofiei

Keep in mind that BUFFER_SCOPE is for DMOs beeing used by the current program, and not all the existing DMOs. I think we should do this:

1. in the begining of core_conversion, collect all DMO (temp and perm) and converted program names in a set
2. expose this set to all downstream rule-sets using a global variable declared in core_conversion
3. in control_flow, if the set contains the 'Action' literal (keep in mind this is case sensitive), then qualify the enum using BlockManager.Action

This will give us a good start for other locations where we would need to disambiguate.

#8 - 03/01/2013 11:07 AM - Costin Savin

- File *cs_upd20130301a.zip* added

Added proposed update. Solution was to parse the dmo_index.xml file and retrieve the existing dmo names in a global HashSet variable which can be used to see if there are any existing dmo with the specified name

#9 - 03/01/2013 11:22 AM - Constantin Asofiei

Please do the following and re-upload:

1. rename dmoUsedNames to usedClassNames (idea is, to save all generated class names which might conflict with P2J class names)
2. collect all class names for the external programs in the usedClassNames, in the core_conversion file
3. in any place where you need to disambiguate (like in control_flow.rules, for "Action"), we will have i.e. usedClassNames.contains("Action") tests.

#10 - 03/01/2013 11:26 AM - Greg Shah

Please look at the com.goldencode.p2j.schema.P2OAccessWorker as a better solution than manually parsing the dmo_index.xml. I don't expect the set is needed either, because we can just lookup a specific name directly using that worker. I should have picked up on this yesterday and mentioned it when Constantin suggested keeping a set.

#11 - 03/01/2013 11:31 AM - Constantin Asofiei

Greg, do we have a way of knowing if a certain name is for a converted class name (associated with the external program)? The set idea was to have all thi in a central repository, so it will be easier to lookup a name for future cases. With your approach, it will be better to add a function in common-progress.rules which returns true if the name is either a DMO name or an external program's class name.

#12 - 03/01/2013 11:58 AM - Costin Savin

- File *cs_upd20130301b.zip* added

Added proposed update

#13 - 03/01/2013 12:06 PM - Eric Faulhaber

Here, give this a try. Register P2OAccessWorker in TRPL and call getAllDMOInterfaceNames() to return a set of all DMO interface names (unqualified) from all databases known to the project.

I haven't tested it. Let me know if you have problems.

#14 - 03/01/2013 12:07 PM - Eric Faulhaber

- File *ecf_upd20130301a.zip* added

...and the attachment.

#15 - 03/01/2013 12:09 PM - Eric Faulhaber

BTW, if this is called more than once, we will want to optimize it, because it copies large sets of strings.

#16 - 03/01/2013 12:14 PM - Costin Savin

I used isDMOInterface(String) in the defined function usedClassName (core_conversion.xml) from the proposed update to check a class name for ambiguities with existing classes. Is that approach not ok?

#17 - 03/01/2013 12:20 PM - Eric Faulhaber

I haven't reviewed your update, but yes, that approach is OK. I was responding to Constantin's idea about maintaining a set of all converted names, DMO or otherwise. If you don't need it, that's fine. If you do, let me know, I just realized there's a bug that I would want to fix.

#18 - 03/01/2013 02:10 PM - Greg Shah

Code Feedback:

In regard to the converted class names, there are several problems that have to be addressed:

1. The approach in core_conversion.xml will only check against classnames of files that have already been processed, not against all files (except for the last program checked).
2. We only need to disambiguate against classnames in our own package, right? As far as I know, we no longer import other packages (or we should not) because all of our RUN statements are dynamically dispatched by the runtime.

Again, I wish I had paid more attention to this earlier. We have the NameMappingWorker that provides TRPL methods that can access the .../p2j/util/SourceNameMapper class. That class has full access to the name_map.xml, which can be used to check converted Java names. The SourceNameMapper.convertjavaProg() is possibly useful in this case. By putting the current package on the front, that method could be called to check if there was a conflicting class in the current package. Note that NameMappingWorker would need an addition to expose that method (or any other method that was added to SourceNameMapper and needs to be used).

Then a simple common-progress.rules function can be added called "classNameConflict" and it can test both the converted program names in the current package and the DMOs.

#19 - 03/04/2013 11:32 AM - Costin Savin

- File `cs_upd20130304a.zip` added

Added the proposed update

#20 - 03/04/2013 07:12 PM - Greg Shah

Code Review:

1. Please merge up to the latest bzd level.

2. This method:

```
public boolean classNameExists(String pkg, String name)
{
    boolean found = false;
    name = pkg + "." + name;
    String progName = SourceNameMapper.convertJavaProg(pkgroot, name);
    if (progName != null)
    {
        found = true;
    }
    return found;
}
```

Would be cleaner like this:

```
public boolean classNameExists(String pkg, String name)
{
    name = pkg + "." + name;
    return (SourceNameMapper.convertJavaProg(pkgroot, name) != null);
}
```

Otherwise this is fine.

#21 - 03/05/2013 05:07 AM - Costin Savin

- File *cs_upd20130305a.zip* added

Added proposed update

#22 - 03/05/2013 09:37 AM - Costin Savin

- File *cs_upd20130305b.zip* added

merge

#23 - 03/05/2013 11:08 AM - Greg Shah

The code looks good. Constantin will be checking in 1 more version of common-progress. Merge with that and we will get this tested and checked in next.

#24 - 03/05/2013 11:27 AM - Costin Savin

- File *cs_upd20130305c.zip* added

merge

#25 - 03/05/2013 11:31 AM - Costin Savin

- File deleted (*cs_upd20130305c.zip*)

#26 - 03/05/2013 11:32 AM - Costin Savin

- File *cs_upd20130305c.zip* added

Copied a wrong update deleted wrong file and uploaded the correct one

#27 - 03/05/2013 11:41 AM - Greg Shah

I'm conversion testing this now.

#28 - 03/05/2013 12:23 PM - Constantin Asofiei

Costin, please double check this, because I think we need to disambiguate this in the opposite direction too, as I think "Action.Buf" reference for a DMO will not work if we statically import BlockManager.* (there is a compile failure related to this in one of the folders)

#29 - 03/05/2013 01:11 PM - Costin Savin

This is a problem indeed, the solution would be to explicitly import the DMO class.

#30 - 03/05/2013 02:05 PM - Costin Savin

- File *cs_upd20130305d.zip* added

Added proposed update

#31 - 03/05/2013 02:32 PM - Greg Shah

cs_upd20130305c.zip has passed conversion testing. Check it in and distribute it. I will look at *cs_upd20130305d.zip* next.

#32 - 03/05/2013 02:34 PM - Greg Shah

In *cs_upd20130305d.zip*, please remove the unchanged files from *cs_upd20130305c.zip* and add a history entry to *control_flow.rules*. Then upload this new update. I will conversion test that next.

#33 - 03/05/2013 02:55 PM - Costin Savin

Committed cs_upd20130305c.zip to bzz as revision number 10243.

#34 - 03/05/2013 02:55 PM - Costin Savin

- File deleted (cs_upd20130305d.zip)

#35 - 03/05/2013 02:56 PM - Costin Savin

- File cs_upd20130305d.zip added

Added cs_upd20130305d.zip only with changes from previous update

#36 - 03/05/2013 03:05 PM - Greg Shah

- Subject changed from *BlockManager Action desambiguation* to *BlockManager Action disambiguation*

I'm conversion testing this now.

#37 - 03/05/2013 04:35 PM - Greg Shah

- Target version set to *Milestone 4*

- Status changed from *WIP* to *Closed*

Passed conversion testing and checked into bzz as revision 10244.

#38 - 03/06/2013 06:50 AM - Constantin Asofiei

0305d.zip doesn't fully solve the issue, as the package name is computed wrong. It uses the package from the nearest BUFFER_SCOPE node, which is not necessary for a dmo in the same schema as the Action dmo.

#39 - 03/06/2013 07:11 AM - Costin Savin

- File cs_upd20130306a.zip added

Added fix update

#40 - 03/06/2013 07:35 AM - Constantin Asofiei

Please remove the dangling text from the 421 and the other edits I mentioned, and re-upload. I'm putting it through conversion regression testing now.

#41 - 03/06/2013 07:36 AM - Costin Savin

- File cs_upd20130306b.zip added

Added update

#42 - 03/06/2013 08:35 AM - Constantin Asofiei

The update has passed conversion regression testing, go ahead and distribute it.

#43 - 03/06/2013 08:57 AM - Costin Savin

Committed to bzz as revision number 10254.

#44 - 03/07/2013 05:28 AM - Costin Savin

- File cs_upd20130307a.zip added

Added update to wrong task deleting it

#45 - 03/07/2013 06:05 AM - Costin Savin

- File deleted (cs_upd20130307a.zip)

#46 - 03/23/2013 03:18 PM - Constantin Asofiei

Converting the extent_params_proc.p test on windows fails with this:

```
.\extent_params_proc.p
java.lang.StringIndexOutOfBoundsException: String index out of range: 1
    at java.lang.String.charAt (String.java:658)
    at java.util.regex.Matcher.appendReplacement (Matcher.java:762)
    at java.util.regex.Matcher.replaceAll (Matcher.java:906)
    at java.lang.String.replaceAll (String.java:2162)
    at com.goldencode.p2j.util.SourceNameMapper.initMappingData (SourceNameMapper.java:832)
    at com.goldencode.p2j.util.SourceNameMapper.convertJavaProg (SourceNameMapper.java:173)
    at com.goldencode.p2j.util.SourceNameMapper.convertJavaProg (SourceNameMapper.java:192)
    at com.goldencode.p2j.convert.NameMappingWorker$Library.classNameExists (NameMappingWorker.java:336)
    at com.goldencode.expr.CE4980.execute (Unknown Source)
    at com.goldencode.expr.Expression.execute (Expression.java:341)
    at com.goldencode.p2j.pattern.NamedFunction.execute (NamedFunction.java:387)
    at com.goldencode.p2j.pattern.AstSymbolResolver.execute (AstSymbolResolver.java:638)
    at com.goldencode.p2j.pattern.CommonAstSupport$Library.execLib (CommonAstSupport.java:1057)
    at com.goldencode.p2j.pattern.CommonAstSupport$Library.evalLib (CommonAstSupport.java:1033)
    at com.goldencode.expr.CE4979.execute (Unknown Source)
    at com.goldencode.expr.Expression.execute (Expression.java:341)
    at com.goldencode.p2j.pattern.Rule.apply (Rule.java:401)
    at com.goldencode.p2j.pattern.Rule.executeActions (Rule.java:640)
    at com.goldencode.p2j.pattern.Rule.coreProcessing (Rule.java:609)
    at com.goldencode.p2j.pattern.Rule.apply (Rule.java:440)
    at com.goldencode.p2j.pattern.RuleContainer.apply (RuleContainer.java:531)
    at com.goldencode.p2j.pattern.RuleSet.apply (RuleSet.java:1)
    at com.goldencode.p2j.pattern.RuleContainer.apply (RuleContainer.java:531)
    at com.goldencode.p2j.pattern.RuleSet.apply (RuleSet.java:1)
    at com.goldencode.p2j.pattern.AstWalker.walk (AstWalker.java:213)
    at com.goldencode.p2j.pattern.AstWalker.walk (AstWalker.java:160)
    at com.goldencode.p2j.pattern.PatternEngine.apply (PatternEngine.java:1119)
    at com.goldencode.p2j.pattern.PatternEngine.processAst (PatternEngine.java:1017)
    at com.goldencode.p2j.pattern.PatternEngine.run (PatternEngine.java:704)
    at com.goldencode.p2j.convert.ConversionDriver.processTrees (ConversionDriver.java:948)
    at com.goldencode.p2j.convert.ConversionDriver.back (ConversionDriver.java:852)
    at com.goldencode.p2j.convert.ConversionDriver.main (ConversionDriver.java:1729)
EXPRESSION EXECUTION ERROR:
-----
nmap.classNameExists(pkgName, className) or p2o.isDMOInterface(className)
^ { Unable to load name mapping data. }
-----
EXPRESSION EXECUTION ERROR:
-----
evalLib("classNameConflict", basePkg, "Action")
^ { Expression execution error @1:6 }
-----
ERROR:
java.lang.RuntimeException: ERROR! Active Rule:
-----
RULE REPORT
-----
Rule Type : WALK
Source AST : [ leave ] BLOCK/STATEMENT/KW_IF/KW_THEN/INNER_BLOCK/BLOCK/INNER_BLOCK/KW_DO/KW_ON/KW_LEAVE/ @52:1
9 {2190433321420}
Copy AST : [ leave ] BLOCK/STATEMENT/KW_IF/KW_THEN/INNER_BLOCK/BLOCK/INNER_BLOCK/KW_DO/KW_ON/KW_LEAVE/ @52:1
9 {2190433321420}
Condition : evalLib("classNameConflict", basePkg, "Action")
Loop : false
```

Files

cs_upd20130228a.zip	21.2 KB	02/28/2013	Costin Savin
cs_upd20130301a.zip	65.5 KB	03/01/2013	Costin Savin
cs_upd20130301b.zip	24 KB	03/01/2013	Costin Savin
ecf_upd20130301a.zip	12.6 KB	03/01/2013	Eric Faulhaber
cs_upd20130304a.zip	79.9 KB	03/04/2013	Costin Savin
cs_upd20130305a.zip	80 KB	03/05/2013	Costin Savin
cs_upd20130305b.zip	80 KB	03/05/2013	Costin Savin
cs_upd20130305c.zip	80 KB	03/05/2013	Costin Savin
cs_upd20130305d.zip	21.4 KB	03/05/2013	Costin Savin
cs_upd20130306a.zip	21.5 KB	03/06/2013	Costin Savin
cs_upd20130306b.zip	21.5 KB	03/06/2013	Costin Savin