Conversion Tools - Bug #6082

automatically add to conversion list all non-skeleton .cls dependencies

02/21/2022 08:29 AM - Constantin Asofiei

Status:	New	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:			
billable:	No	**case_num:**	
vendor_id:	GCD	**version:**	

Description

Related issues:

Related to Conversion Tools - Feature #5630: add missing classes referenced d...	**New**
Related to Conversion Tools - Feature #7169: drive conversion order using use...	**New**
Related to Conversion Tools - Feature #7180: create a centralized manager for...	**New**

History

#1 - 02/21/2022 08:31 AM - Constantin Asofiei

During conversion, if a .cls or a .p has dependencies to non-skeleton classes which are not included in the initial conversion list, the conversion will fail at some point in the annotations or main conversion phase.

The parsing phase can be improved to recursively include all .cls files (loaded during parsing phase and which are not part of the conversion list), until no more new .cls files are loaded.

#2 - 02/21/2022 08:34 AM - Greg Shah

- Related to Feature #5630: add missing classes referenced during parsing to the conversion list in file list mode added

#3 - 02/21/2022 08:35 AM - Greg Shah

This may be a duplicate of #5630.

#4 - 05/11/2022 11:07 AM - Constantin Asofiei

In #6351-10 Greg Shah wrote:

> Sub-classes should be re-converted every time a parent class is converted.

This is a real issue, as the converted Java method name in sub-classes must match the one in the parent class.

Of course, on each step this will require to go back and check for all references/sub-classes/etc until none are left.

#6 - 05/01/2023 07:47 AM - Constantin Asofiei

Greg, the changes which helped me get through the 62k files for #7199 are in 6082a rev 14553 (created from trunk rev 14552). I don't really like the approach (and SymbolResolver.parseHierarchy may lose .cls dependencies), but I just wanted to archive these changes, so they do not get lost. It also breaks (again) concurrent conversion or parsing.

#7 - 05/02/2023 09:49 AM - Greg Shah

- Related to Feature #7169: drive conversion order using user-specified dependencies added

#8 - 05/02/2023 09:49 AM - Greg Shah

- Related to Feature #7180: create a centralized manager for the conversion list and per-file/project status and logging added

#9 - 05/02/2023 09:49 AM - Greg Shah

I've looked at the changes. Yes, you are right that they are not something we could pull into trunk as is. I had similar code to this and had to remove it because it did not result in the correct list. I went in favor of the current wa.processed approach used during pre-scan which was also nasty but I did get it working reliably.

This area really needs to be reworked with #7169 and #7180.