Non-Conformances - Bug #4220

Exposed visibility of xsl:param

2019-05-15 12:39 - Michael Kay

Status:	New	**Start date:**	2019-05-15
Priority:	Normal	**Due date:**	
Assignee:	Michael Kay	**% Done:**	0%
Category:		**Estimated time:**	0:00 hour
Sprint/Milestone:			
Applies to branch:		**Found in version:**	
Fix Committed on Branch:		**Fixed in version:**	
Fixed in Maintenance Release:		**Platforms:**	

Description

Unit test s9apitests/TestPackage/testPackageRenamingExported is failing with an error saying it cannot accept xsl:param elements with private visibility as hidden.

The spec says (a) that xsl:param elements are always public, and (b) that their visibility cannot be changed by xsl:expose or xsl:accept.

What this test is trying to do is to compile a single package three times with different settings of a static parameter, and then import all three exported packages into a single top-level package (using package aliases to achieve this). It might be that we won't be able to get this to work, but we need to investigate why it's failing the way that it is. The spec says that xsl:param is public, but we're reporting it as private.

History

#1 - 2019-05-15 12:55 - Michael Kay

There's an inconsistency in the spec here.

§3.5.3.1 () says: (a) In the case of an xsl:param element there is no explicit visibility attribute; rather the declaration has the implicit attribute visibility="public". (b) The visibility of a variable declared using an xsl:param element is always public. No xsl:expose element ever matches an xsl:param component.

§9.6 (Static parameters) says: When the static attribute [of xsl:param] is present with the value yes, the visibility attribute must not have a value other than private.

In fact, xsl:param does not allow a visibility attribute.

As far as I can see this hasn't been picked up in any existing bug report on the spec.

#2 - 2019-05-15 13:03 - Michael Kay

Saxon is giving static xsl:param components a visibility of PRIVATE, and non-static xsl:param components a visibility of PUBLIC, which seems a reasonable interpretation of the spec.

I don't think we're doing anything to stop xsl:expose and xsl:accept changing this.

#3 - 2019-05-15 13:06 - Michael Kay

As regards the JUnit tests, if I change pack-120-top.xsl to remove the xsl:accept declarations, I now get "duplicate variable opname" - but if the variable is private in both used packages, this should not happen. In other words, we get different results depending on whether the used packages are exported (saved as SEF files) or not.

#4 - 2020-01-23 19:41 - Michael Kay

Status report: the unit test is still failing (on the development branch, at least)

#5 - 2020-02-26 13:31 - Michael Kay

- Project changed from Saxon to Non-Conformances

- Category deleted (XSLT 3.0 packages)