

Saxon - Bug #4839

collection(): failed to parse XML file

2020-11-25 19:40 - O'Neil Delpratt

Status: Closed	Start date: 2020-11-25
Priority: Normal	Due date:
Assignee: O'Neil Delpratt	% Done: 100%
Category: .NET API	Estimated time: 0:00 hour
Sprint/Milestone:	Spent time: 0:00 hour
Legacy ID:	Fix Committed on Branch: 10, trunk
Applies to branch: 10, trunk	Fixed in Maintenance Release: 10.5
Description	
Reported by user on saxon mailing list: When the collection() feature is used in the XSLT stylesheet. The following error occurs: <code>collection(): failed to parse XML file file:/D:/Test/sample1.xml: I/O error reported by XML parser processing file:/D:/Test/sample1.xml: Could not find file 'D:\Test\doctype.dtd'.</code> It appears the default CollectionFinder does not use XsltTransformer.InputXmlResolver; The problem is the C# .NET code does not have its own CollectionFinder as in the Java product	

History

#1 - 2020-11-26 07:44 - Emanuel Wlaschitz

Thanks for logging this!

As far as I can tell, the CollectionFinder *does* exist and we can set one using `processor.SetProperty(Feature<CollectionFinder>.COLLECTION_FINDER, new CustomCollectionFinder())` (or using `transformer.Implementation.setCollectionFinder(new CustomCollectionFinder())` to be more localized) and its `findCollection` method will be called.

I just don't see how we could affect how documents are loaded, as the documentation seems to suggest it only returns URIs and not loaded documents.

#2 - 2020-11-26 09:46 - Michael Kay

Sorry to confuse. What I meant to say was that the default in the .NET product is to use the Java CollectionFinder, which of course has no knowledge of .NET-specifics like the XmlResolver. (I'm not even sure if that statement is correct, it needs further investigation).

#3 - 2020-11-26 11:35 - O'Neil Delpratt

- Tracker changed from Bug to Feature

#4 - 2021-01-06 18:57 - O'Neil Delpratt

- Status changed from New to In Progress

- Applies to branch deleted (9.9)

I have added the `@CollectionFinder@` feature to .NET, which will be available in the next maintenance release.

Users can now define their own `@ICollectionFinder@` and set it on the Processor object to be used in XQuery, XPath XSLT APIs.

As in Java, we now have `@IResourceCollection@` interface to map URI of the collection into a sequence of Resource objects. We have a number of implementations of `@IResourceCollection@` available for users to use: `@CatalogCollection@`, `@JarCollection@` and `@DirectoryCollection@`.

NUnit tests added.

I am leaving this bug issue open until API doc is complete.

#5 - 2021-01-07 01:12 - O'Neil Delpratt

- Status changed from In Progress to Resolved
- % Done changed from 0 to 100
- Fix Committed on Branch 10, trunk added

Bug fixed and committed on Saxon10 and trunk branches.

#6 - 2021-01-11 13:06 - Emanuel Wlaschitz

Just checking, this `CollectionFinder` will allow us to set a custom `.NET XmlResolver` to be used when loading individual entries of the collection, right?

#7 - 2021-01-11 14:05 - O'Neil Delpratt

Yes, the `CollectionFinder` should use your custom `XsltTransformer.InputXmlResolver`. If you have a sample application I will happy to test your setup with this new feature.

#8 - 2021-01-11 15:15 - Emanuel Wlaschitz

We don't have a ready-to-use sample application, but we can make one.

Since we don't have the change yet, we ran the following with `transform.exe collection.xml -it:main -` but I'm confident you can turn this into a `.NET` testcase:

`collection.xml` (which simply looks at all XML files in the same folder and prints their root element name)

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" xmlns:xs="http://www.w3.org/2001/XMLSchema"
exclude-result-prefixes="xs" version="2.0">

  <xsl:template name="main">
    <xsl:apply-templates select="collection('./?select=*.xml')" mode="print-root" />
  </xsl:template>

  <xsl:template match="/" mode="print-root">
    <xsl:text>&#xA;</xsl:text>
    <xsl:value-of select="name(/*)" />
  </xsl:template>

</xsl:stylesheet>
```

`a.xml` in same folder as `collection.xml` (the DTD does not really matter, but removing it allows the XSLT to run)

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE test SYSTEM "does-not-exist.dtd">
<a/>
```

You can add more XML files if you want, both with and without DOCTYPE, where the ones with a System ID trigger the exception from this issue.

In C#, we'd use a custom `XmlResolver` like this:

```
public class DtdIgnoringResolver : XmlResolver
{
    private readonly XmlResolver _innerResolver;

    public DtdIgnoringResolver(XmlResolver innerResolver)
    {
        _innerResolver = innerResolver ?? throw new ArgumentNullException(nameof(innerResolver));
    }

    public override ICredentials Credentials
    {
        set { _innerResolver.Credentials = value; }
    }

    public override object GetEntity(Uri absoluteUri, string role, Type ofObjectToReturn)
    {
        if (!string.IsNullOrEmpty(absoluteUri?.OriginalString) && IsDtdOrSchema(absoluteUri.OriginalString))
            return Stream.Null;
        return _innerResolver.GetEntity(absoluteUri, role, ofObjectToReturn);
    }

    private static bool IsDtdOrSchema(string filePath)
    {
        return
            filePath.EndsWith(".dtd", StringComparison.InvariantCultureIgnoreCase) ||

```

```
        filePath.EndsWith(".xsd", StringComparison.InvariantCultureIgnoreCase);  
    }  
}
```

The piece of code we use is a bit more involved and usually applies the XSLT to a file (rather than running a named template), but I'm sure this suffices. If not, let me know.

Thanks again.

#9 - 2021-01-12 17:29 - O'Neil Delpratt

Hi,

Thanks for sending this example, which I used to create a nunit test case. I can confirm it now works as expected. No longer seeing the exception.

#10 - 2021-01-13 15:41 - O'Neil Delpratt

- Status changed from Resolved to In Progress

I am reopening this bug issue because we are still seeing the exception described in the initial report.

There was a bug in the test case in comment #9. I have set the InputXmlResolver to the DtdIgnoringResolver on the Xslt30Transformer, but this is not being filtered through to the CollectionFinder.

Further investigation required.

#11 - 2021-03-30 14:16 - Michael Kay

- Tracker changed from Feature to Bug

#12 - 2021-04-14 17:33 - O'Neil Delpratt

- Status changed from In Progress to Resolved

Bug fixed and test of nunit tests created.

#13 - 2021-04-14 17:49 - O'Neil Delpratt

- Status changed from Resolved to Closed

- Fixed in Maintenance Release 10.5 added

Bug fix applied to Saxon 10.5 maintenance release.