# Defect in comparing deployed dacpac with source containing a fulltext index

## Versions

* Visual Studio 2017 Professional – Version 15.8.5
* .NET Framework Version 4.7.03056
* SQL Server Data Tools Version 15.1.61808.07020

## Scenario #1 – Deploying through visual studio, comparing using schema compare

I have a sql server database project with a filegroup created as FG\_Standard:-

ALTER DATABASE [$(DatabaseName)]

ADD FILEGROUP [FG\_Standard]

I have a full text catalogue created as below:-

CREATE FULLTEXT CATALOG [FullTextCatalog1]

WITH ACCENT\_SENSITIVITY = OFF

AUTHORIZATION [dbo];

I have one table [dbo].[SomeTable], with an ID column a nvarchar for the searchable txt and a PK over the ID which is to be added to FG\_Standard filegroup: -

CREATE TABLE [dbo].[SomeTable]

(

[ID] INT IDENTITY (1, 1) NOT NULL,

[SearchableText] NVARCHAR (200) NOT NULL,

CONSTRAINT [PK\_SomeTable] PRIMARY KEY CLUSTERED ([ID] ASC) WITH (FILLFACTOR = 90)

) ON FG\_Standard

GO

I have a full text index on this table over the searchable text column as below:-

CREATE FULLTEXT INDEX ON [dbo].[SomeTable]

([SearchableText] LANGUAGE 1033)

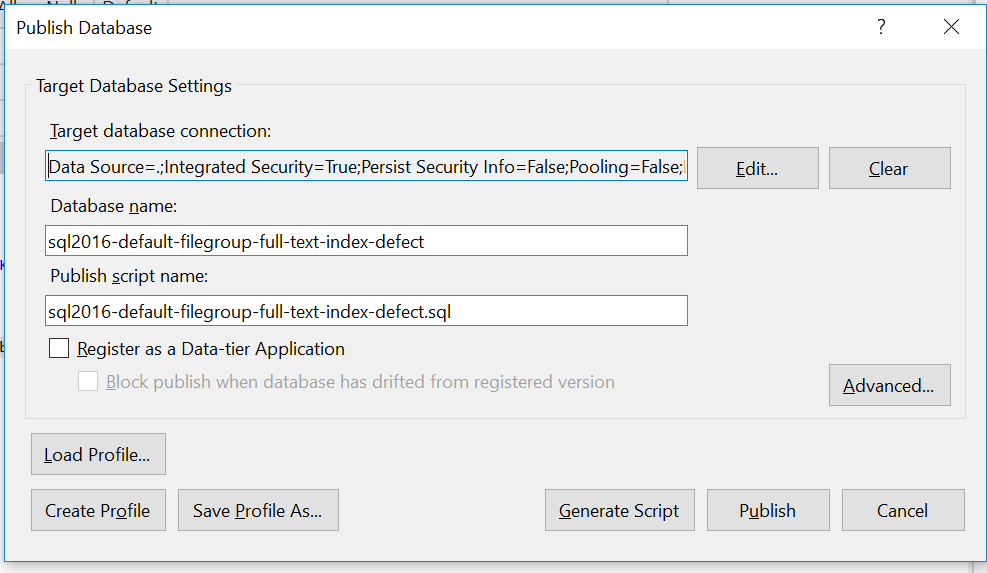
KEY INDEX [PK\_SomeTable]

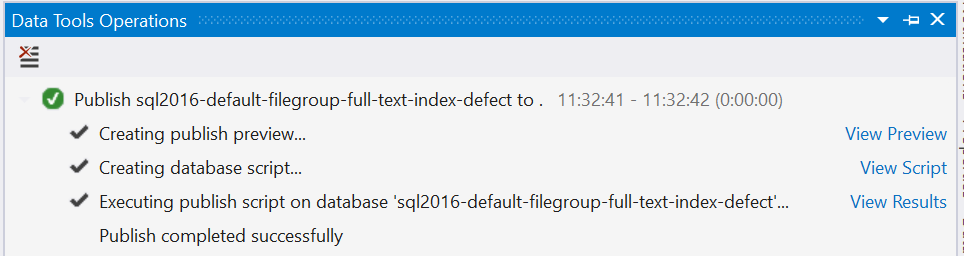
ON [FullTextCatalog1]

WITH STOPLIST OFF;

GO

I can publish this without any issues using the publish tools in visual studio:

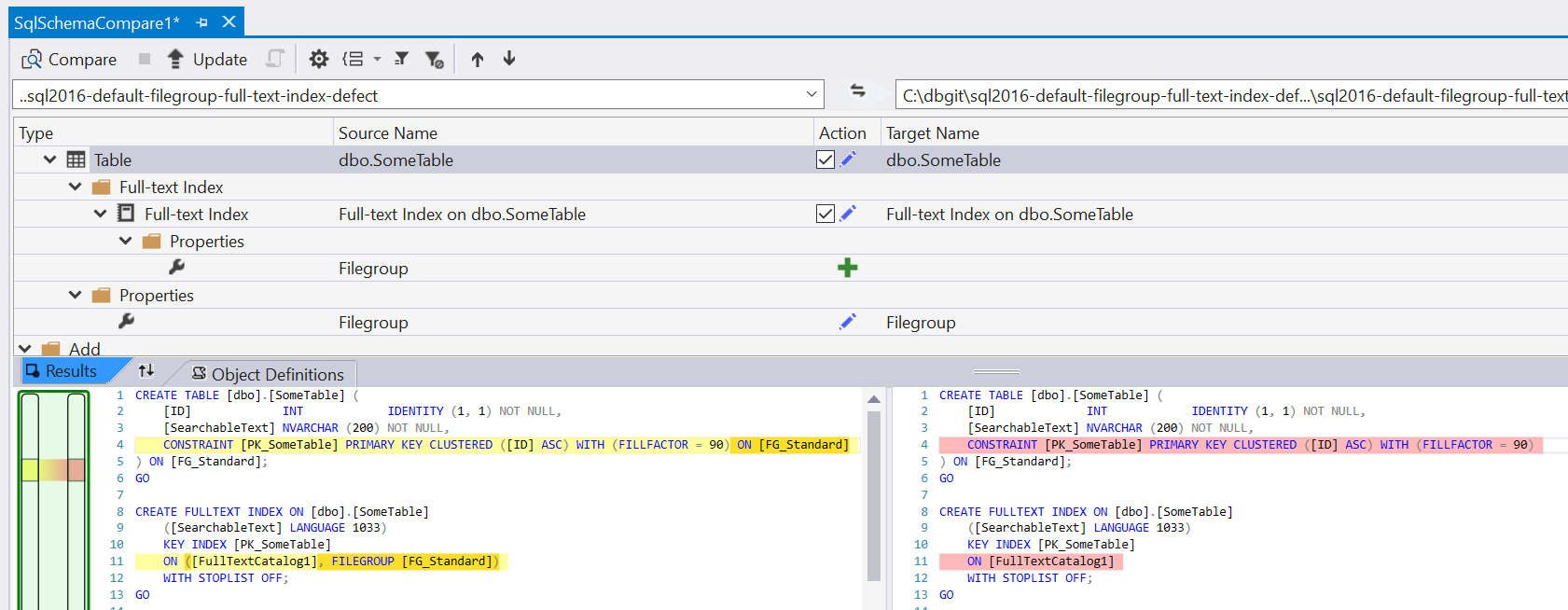




When I use the schema compare app to compare the differences between published db and source the source code I am seeing that the filegroup on the published db has been included in the compare causing differences between the PK and full text catalogue…

Published DB

Source Code



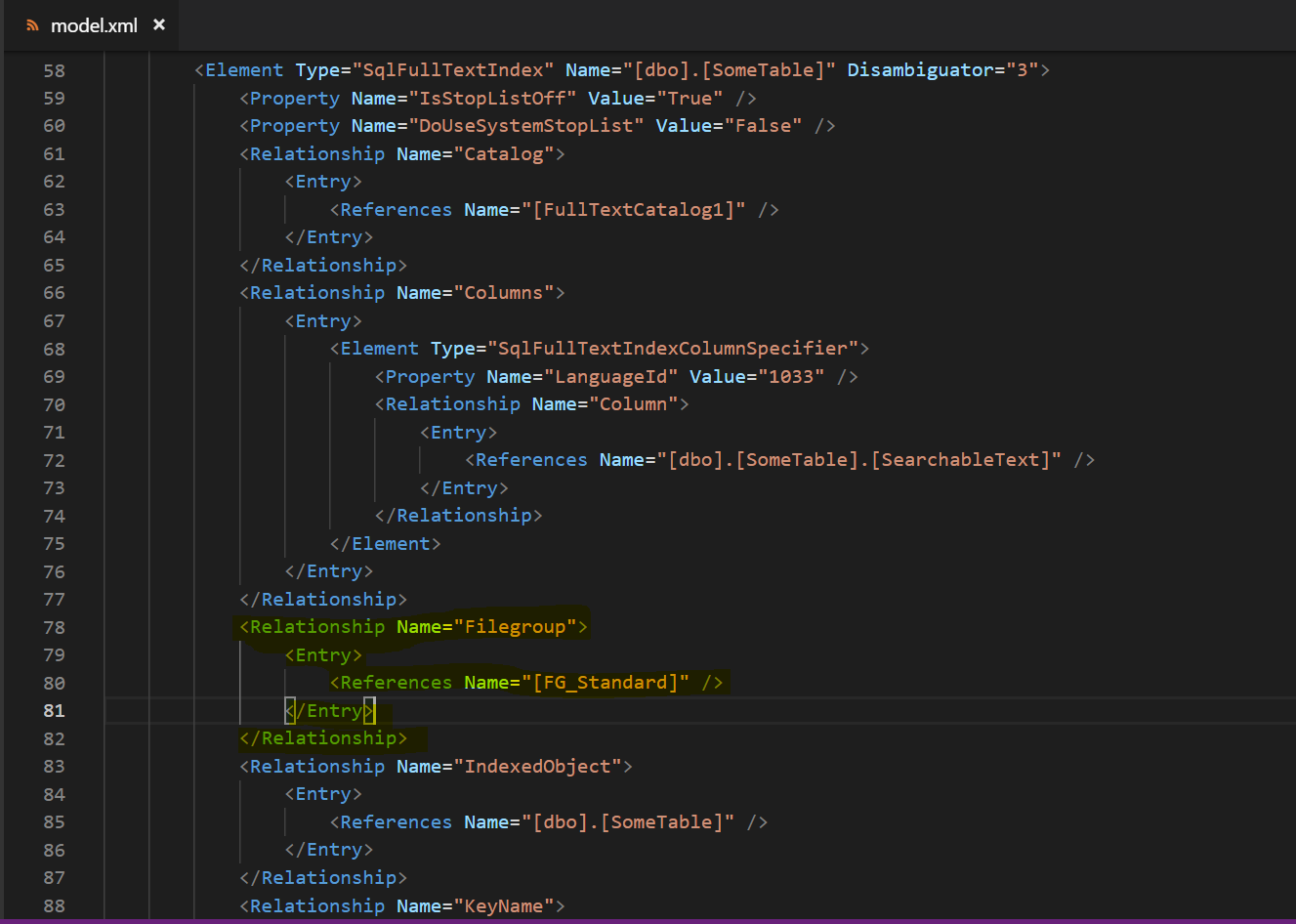
## Scenario #2 – Deploying through visual studio, extracting dacpac and comparing using schema compare

The source code remains the same as scenario #1, and we publish to the server in exactly the same way, however we extract the dacpac from the server using sqlpackage.exe as below:-

"C:\Program Files (x86)\Microsoft Visual Studio\2017\Professional\Common7\IDE\Extensions\Microsoft\SQLDB\DAC\150\sqlpackage.exe" /Action:Extract /OverwriteFiles:True /TargetFile:".\ExtractedDacPac\sql2016-default-filegroup-full-text-index-defect.dacpac" /SourceDatabaseName:sql2016-default-filegroup-full-text-index-defect /SourceServerName:.

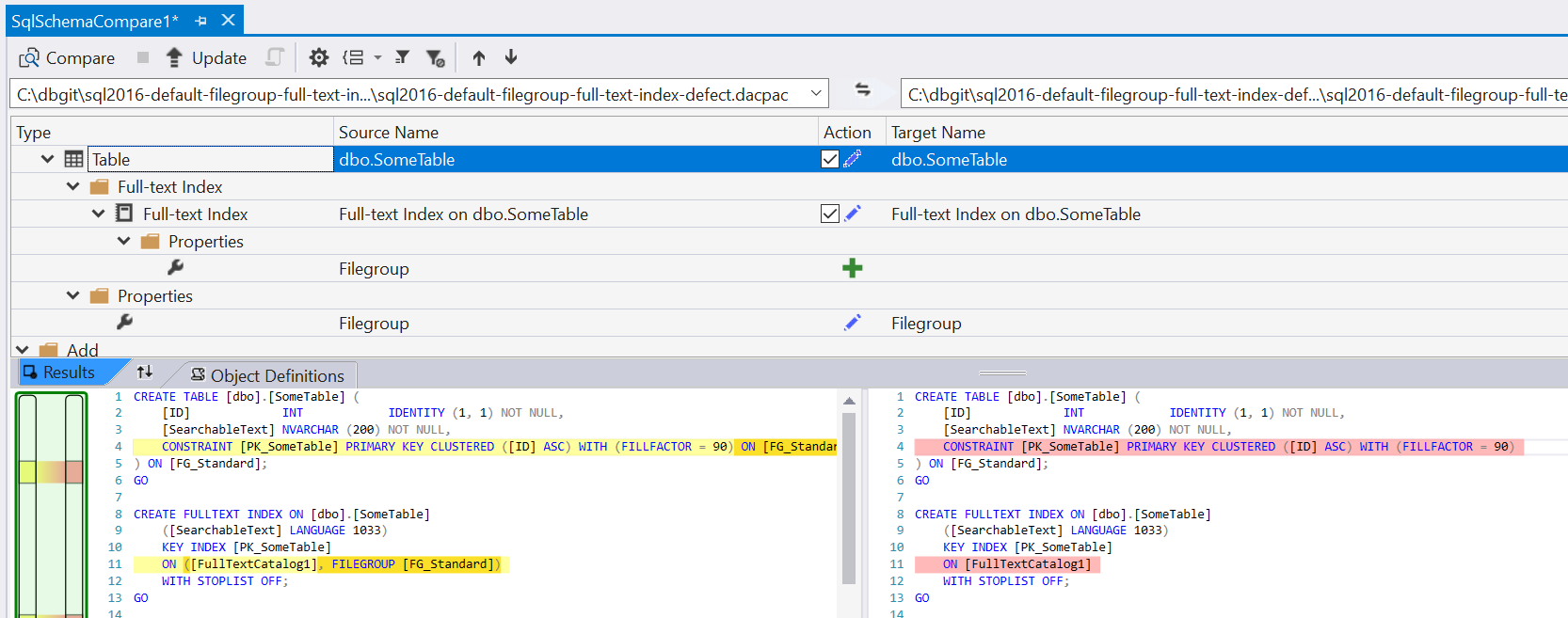
PAUSE

This produces a dacpac, if you extract dacpac with 7zip or any other compression tool and examine the model.xml within it, you will notice that it does infactt contains the FileGroup assigned to the full text index, even though its not defined this way in the above code:-



As you can see between lines 78-82 a FileGroup reference has been extracted from the database.

Therefore when you run the below schema compare between extracted dacpac and the source code you will see the added filegroup reference from the database as well as PK differences:-



Source Code

Extracted Dacpac

## Conclusion

When publishing the project the process is assigning the tables filegroup to the Full-text index and PK as default, which is correct.

However, when using the schema compare to compare the deployed database back to the source code, this should also default to the filegroup of the table if filegroups have not been defined on the objects. I.e.

In my opinion, if I have a table with a filegroup assigned, and I have a PK and Full-text Index on that table without a filegroup assigned, when comparing the filegroup of the objects the tables’ filegroup should be used in the comparison if that object doesn’t have a filegroup assigned. Which is not what is currently happening when using the schema compare.

## Github.com Repo

I have added the defect to github.com for your convenience so that you can run through the scenarios:

<https://github.com/fenngineering/sql2016-nullable-computed-column-defect>

Thanks

Andrew