Article
Peter Steiwer · Jan 10, 2020   4m read

Understanding Missing Relationship Build Errors in InterSystems IRIS Business Intelligence

When using Related Cubes in InterSystems IRIS BI, cubes must be built in the proper order. The One side must be built before the Many side. This is because during build time for the Many side, it looks up the record on the One side and creates a link. If the referenced record is not found on the One side, a Missing Relationship build error is generated. The One side is going to be the independent side of the relationship, AKA the side of the relationship that is referenced by the Many side or the Dependent cube. For example: Patients contain a reference to their Doctor. The Doctor does not contain references to each of their Patients. Doctors is the One, or Independent side. Patients is the Many, or Dependent side. For more information about setting up Cube Relationships, please see the documentation.

WARNING: If you rebuild the One side without rebuilding the Many side, the Many side may point to the wrong record. It is not guaranteed that a record in your cube will always have the same ID. The relationship link that is created is based on ID. YOU MUST REBUILD THE MANY SIDE AFTER BUILDING THE ONE SIDE.

To ensure your cubes are always built in the proper order, you can use the Cube Manager.

When debugging Build Errors, please also debug them in the Build Order. This is because errors can cascade and you don't want to spend time debugging an error just to find out it is because a different error happened first.

Understanding the Missing Relationship Build Error Message

```
SAMPLES>do ##class(%DeepSee.Utils).%PrintBuildErrors("RELATEDCUBES/PATIENTS")
   1   Source ID: 1
       Time: 01/03/2020 15:30:42
       ERROR #5001: Missing relationship reference in RelatedCubes/Patients: source ID 1 missing reference to RxPrimaryCarePhysician 1744
```

Here is an example of what the Missing relationship build error looks like. We will extract some of these values from the message to understand what is happening.

Missing relationship reference in [Source Cube]: source ID [Source ID] missing reference to [Related Cube Reference] [Related Source ID]

In our error message, we have the following values:

Source Cube = RelatedCubes/Patients

Source ID = 1

Related Cube Reference = RxPrimaryCarePhysician

Related Source ID = 1744

Most of these are pretty straightforward except for the Related Cube Reference. Sometimes the name is obvious, other times it is not. Either way, we can do a little bit of work to find the cube this reference.

Step 1) Find the Fact Class for the Source Cube.

```
SAMPLES>w ##class(%DeepSee.Utils).%GetCubeFactClass("RelatedCubes/Patients")
BI.Model.RelCubes.RPatients.Fact
```

Step 2) Run an SQL query to get the Fact class the Related Cube Reference is pointing to:

```
SELECT Type FROM %Dictionary.PropertyDefinition where ID='[Source Cube Fact Class]||[
Related Cube Reference]'
```

example:

```
SELECT Type FROM %Dictionary.PropertyDefinition where ID='BI.Model.RelCubes.RPatients
.Fact||RxPrimaryCarePhysician'
```

Which returns a value of: BI.Model.RelCubes.RDoctors.Fact

Step 3) Now that we have the Related Cube Fact Class, we can run an SQL query to see if this Related Source ID does not have an associated fact in our Related Cube Fact Table.

```
SELECT * FROM BI_Model_RelCubes_RDoctors.Fact WHERE %SourceId=1744
```

Please note that we had to use the SQL table name instead of the class name here. This can typically be done by replacing all "." excluding ".Fact" with "_".

In this case, 0 rows were returned. This means it is still the case that the required related fact does not exist in the related cube. Sometimes after spending the time to get to this point, a synchronize may have happened to pull this new data in. At this point, the Build Error may no longer be true, but it has not yet been cleared out of the Build Errors global. Regular synchronization does not clean entries in this global that have been fixed. The only way to clean the Build Errors global is to run a Build against the cube OR running the following method:

```
Do ##class(%DeepSee.Utils).%FixBuildErrors("CUBE NAME WITH ERRORS")
```

If we now had data for the previous SQL query, the %FixBuildErrors method should fix the record and clear the error.

Step 4) Since we do not have this record in our Related Cube Fact Table, we should check the Related Cube Source Table to see if the record exists. First we have to find the Related Source Class by viewing the SOURCECLASS parameter of the Related Cube Fact Class:

```
SAMPLES>w ##class(BI.Model.RelCubes.RDoctors.Fact).#SOURCECLASS
BI.Study.Doctor
```

Step 5) Now that we have the Related Source Class, we can query the Related Source Table to see if the Related Source ID exists:

```
SELECT * FROM BI_Study.Doctor WHERE %ID=1744
```

If this query returns results, you should determine why this record does not exist in the Related Cube Fact Table. This could simply be because it has not yet synchronized. It could also have gotten an Error while building this fact. If this is the case, you need to remember to diagnose all Build Errors in the proper Build Order. It can often be the case that lots of errors cascade from one error.

If this query does not return results, you should determine why this record is missing from the Related Source Table. Perhaps some records have been deleted on the One side but records on the Many side have not yet been reassigned or deleted. Perhaps the Cube Relationship is configured incorrectly and the Related Source ID is not the correct value and the Cube Relationship definition should be changed.

This guide is a good place to start, but please feel free to contact the WRC. The WRC can help debug and diagnose this with you.

#Analytics #Caché #InterSystems IRIS #InterSystems IRIS BI (DeepSee)

Source URL:https://community.intersystems.com/post/understanding-missing-relationship-build-errors-intersystems-iris-business-intelligence