Article
Alessandro Marin · Nov 14, 2016 5m read

DeepSee – Less records built than rows in source table

The DeepSee TroubleShooting Guide helps you track down and fix problems occurring in your DeepSee project. A common problem is finding less records than expected in a DeepSee Cube or a related Subject Area. The DeepSee TroubleShooting Guide suggests starting your investigation by checking the following:

Check cube for build restrictions

Check if maxFacts is used

Check if Build Errors are occurring

Check cube for build restrictions

Build restrictions can be implemented in two ways:

- By specifying the Build restriction option in Architect > click on the cube/SA definition. This option will set the buildRestriction attribute in the Cube class visible from Studio.

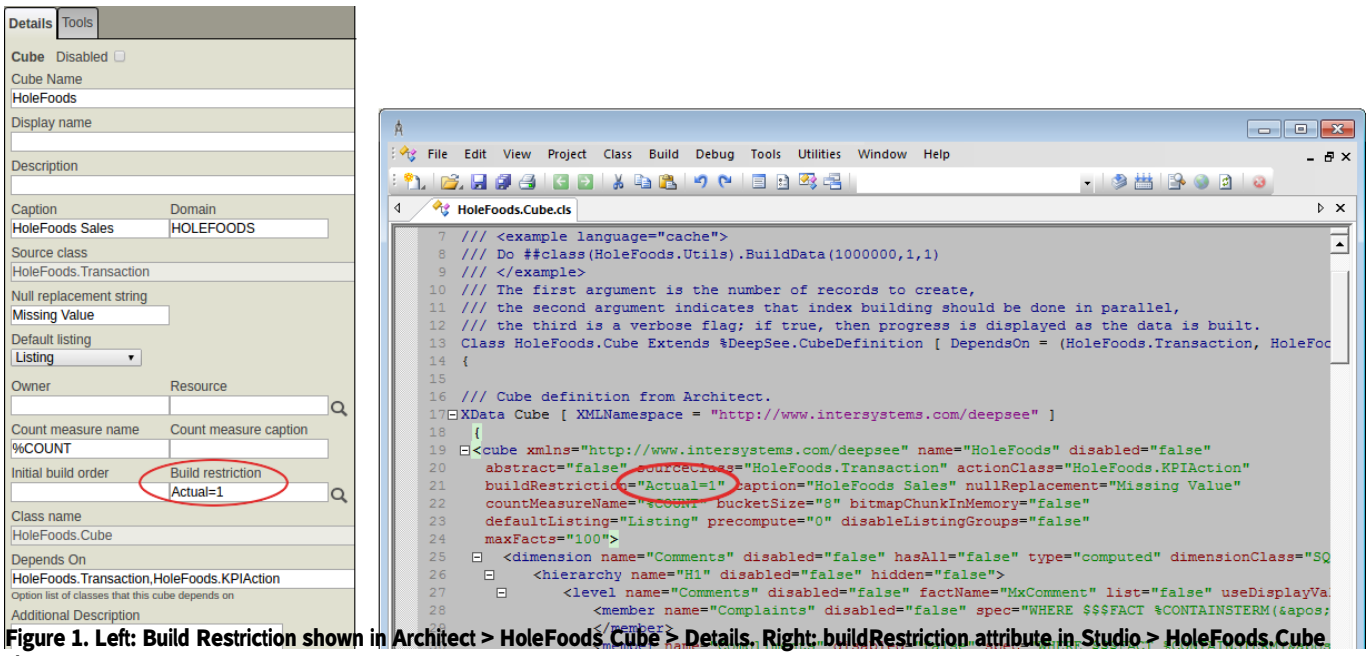

Figure 1. Left: Build Restriction shown in Architect > HoleFoods.Cube > Details. Right: buildRestriction attribute in Studio > HoleFoods.Cube class.

- By implementing the %OnProcessFact. This method can be used to skip building facts based on either the fact ID, the level or measure name (1) in the fact table, or if the fact is a new row. For example, the following code does not build facts with ID lower than 100 that have an amount of sale lower than 2:

```
ClassMethod %OnProcessFact(pID As %String, ByRef pFacts As %String,
Output pSkip As %Boolean, pInsert As %Boolean) As %Status {
If ((pID<100) && (pFacts("MxAmountOfSale")<2)) {
Set pSkip=1
} Else {
Set pSkip=0
}
Quit $$$OK
}
```

Check if maxFacts is used

The maxFacts attribute can be used in the cube definition to limit the number of facts to be built. Notice that in Architect maxFacts will appear in the popup window that opens when you Build the cube, but will not be overridden by the UI. The maxFacts attribute is a useful tool in the development phase but should be removed before deploying the cube.

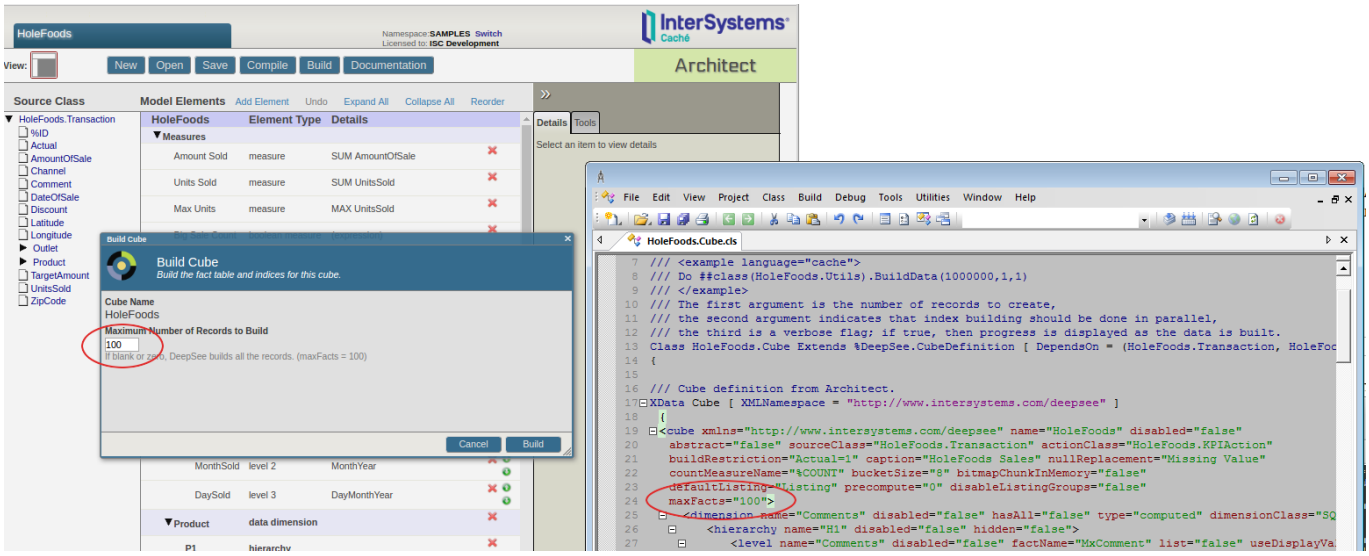

Figure 2. Left: maxFacts in Architect > Build popup window. Right: maxFacts attribute in Studio > HoleFoods.Cube class.

Also note that when you build your cube programmatically the fifth argument to the %DeepSee.Utils:%BuildCube method specifies the maximum number of rows in the fact table, for example:

```
SAMPLES> Do ##class(%DeepSee.Utils).%BuildCube("HoleFoods", , , , 100)
```

Check if Build Errors are occurring

Build errors might take place when a cube is built. The example in the picture below shows build errors as generated when building from Architect or using the %DeepSee.Utils:%BuildCube method.

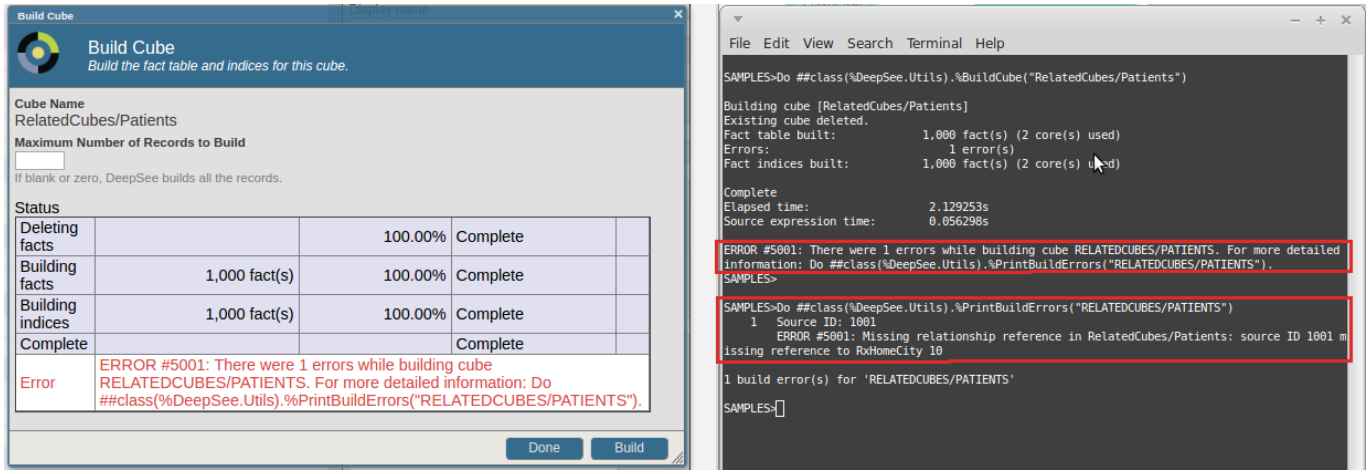

Figure 3. Left: Build errors in Architect > Build popup window. Right: build errors and %PrintBuildErrors when building a cube from terminal.

More information about the errors can be found in the ^DeepSee.BuildErrors(<cubename>) global or by using the %DeepSee.Utils:%PrintBuildErrors. In the example above, a missing reference error was present:

```
SAMPLES>Do
##class(%DeepSee.Utils).%PrintBuildErrors("RelatedCubes/Patients")
  1 Source ID: 1001
  ERROR #5001: Missing relationship reference in
RelatedCubes/Patients: source ID 1001 missing reference to RxHomeCity
10
1 build error(s) for 'RelatedCubes/Patients'
```

A full description of missing reference errors is outside the scope of this post, but an explanation for the error is given in the note below (2).

Check the answer to this post for a detailed example of how to use the % DeepSee.Utils:%PrintBuildErrors method.

Note 1

To find the level or measure name from the fact table I suggest using %DeepSee.Utils.%Analyze from terminal:

```
SAMPLES> Do ##class(%DeepSee.Utils).%Analyze(<cubename>,"i")
```

or the following command:

```
SAMPLES> Do ^%G

^DeepSee.Cubes("cubes",<cubename>,"fact","prop", ,"alias")
```

The Field Names paragraph in the documentation describes in detail how DeepSee determines the names for the measure, level, and dimension fields.

Go back to text

Note 2

Referring to the example in Figure 3:

```
SAMPLES> Do
##class(%DeepSee.Utils).%BuildCube("RelatedCubes/Patients")

Building cube [RelatedCubes/Patients]

Existing cube deleted.

Fact table built: 1,000 fact(s) (2 core(s) used)

Errors: 1 error(s)

Fact indices built: 1,000 fact(s) (2 core(s) used)

Complete

Elapsed time: 2.136473s

Source expression time: 0.055421s

ERROR #5001: There were 1 errors while building cube
RelatedCubes/Patients. For more detailed information: Do
##class(%DeepSee.Utils).%PrintBuildErrors("RelatedCubes/Patients").
```

Building RelatedCubes/Patients generated one error. The output suggested to view the details using the %DeepSee.Utils:%PrintBuildErrors method:

```
SAMPLES> Do
##class(%DeepSee.Utils).%PrintBuildErrors("RelatedCubes/Patients")
```

```
1 Source ID: 1001

ERROR #5001: Missing relationship reference in RelatedCubes/Patients:
source ID 1001 missing reference to RxHomeCity 10

1 build error(s) for 'RelatedCubes/Patients'
```

The output says that the missing reference error took place for the fact with source ID = 1001 in the source table of the RelatedCubes/Patients cube (which is the DeepSee.Study.Patient class).

At this point it is a good idea to investigate the patient record. This can be done in several ways (e.g. by opening the class instance using Cache Objects, or the SQL query utility in Management Portal). Using the %SQL.Statement and %DeepSee.ResultSe classes from terminal:

```
SAMPLES> Set rs = ##class(%SQL.Statement).%ExecDirect(, "SELECT ID,
HomeCity FROM DeepSee_Study.Patient WHERE ID=1001")

SAMPLES> Do rs.%Display()

ID HomeCity

1001 10
```

The output shows that Patient with ID=1001 points to a City with ID=10. As expected, this City is present in the DeepSeeStudy.City source table:

```
SAMPLES> Set rs = ##class(%SQL.Statement).%ExecDirect(, "SELECT * FROM
DeepSee_Study.City WHERE ID=10")

SAMPLES> Do rs.%Display()

ID Name Population PostalCode PrincipalExport

10 Christiania 850 1440 bikes
```

The RelatedCubes/Patients cube defines a relationship with the RelatedCubes/Cities cube. Building RelatedCubes/Patients requires the RelatedCubes/Cities cube to contain all cities referenced by each patient. As we can see, the RelatedCubes/Cities cube does not contain the referenced city Christiania:

```
SAMPLES> Set rs=##class(%DeepSee.ResultSet).%ExecuteDirect("SELECT NON
EMPTY [CitiesD].[H1].[City].Members ON 1 FROM [RELATEDCUBES/CITIES]")

SAMPLES> Do rs.%Print()

1 Cedar Falls 1

2 Centerville 1

3 Cypress 1

4 Elm Heights 1

5 Juniper 1

6 Magnolia 1
```

```
7 Pine 1

8 Redwood 1

9 Spruce 1
```

To solve the error the RelatedCubes/Cities cube must be built first, followed by RelatedCubes/Patients one:
SAMPLES>Do ##class(%DeepSee.Utils).%BuildCube("RelatedCubes/Cities")
Building cube [RelatedCubes/Cities]
Existing cube deleted.
Fact table built: 10 fact(s) (2 core(s) used)
Fact indices built: 10 fact(s) (2 core(s) used)
Complete
Elapsed time: 2.003276s
Source expression time: 0.000003s
SAMPLES>Do ##class(%DeepSee.Utils).%BuildCube("RelatedCubes/Patients")
Building cube [RelatedCubes/Patients]
Existing cube deleted.
Fact table built: 1,001 fact(s) (2 core(s) used)
Fact indices built: 1,001 fact(s) (2 core(s) used)
Complete
Elapsed time: 2.139756s
Source expression time: 0.057484s

Go back to text

#InterSystems IRIS BI (DeepSee) #Terminal

Source URL:https://community.intersystems.com/post/deepsee-%E2%80%93-less-records-built-rows-source-table