
Article

[Peter Steiwer](#) · Nov 26, 2019 3m read

Designing valid hierarchies in DeepSee

When designing a hierarchy in DeepSee, a child member must have only one parent member. In the case where a child corresponds to two parents, the results can become unreliable. In the case where two similar members exist, their keys must be changed so that they are unique. We will take a look at two examples to see when this happens and how to prevent it.

Example 1

There are a handful of states with a city named Boston. In my sample data, I have records from both Boston, MA and Boston, NY. My dimension is defined as:

» ▼ Location	data dimension		✖
H1	hierarchy		✖
State	level 1	State	✖ ⬆ ⬆
City	level 2	City	✖ ⬆ ⬆

City and State are simply strings in my case. When built into my cube, I will have two state members: "MA" and "NY" and two city members: "Boston" and "Boston". Why do I have two Boston members and not just one? Since a member can't have two parent members, we need to create two separate members, one for each parent. Unfortunately, we are now in the "Bad Hierarchy" situation since we have one key that points to two different members.

To fix this, we need to make the key unique. Instead of directly using the "City" property as the source property for the level, we can use a source expression to make this member unique.

Name

City

Display name

Description

Source Values

☐ Property

☒ Expression

%source.City_","_%source.State

This will solve the problem for us, but there may be a side effect here we don't want. With this expression, we see the following results in our pivot table:

State	City	All
MA	Boston,MA	1
NY	Boston,NY	1

This may or may not be an acceptable display format. At this point, our key and member name are the same, we can do a little more work to make our display read just "Boston", but have a unique key behind it. Please see the [documentation](#) for more details.

To recap, different members are required to have unique keys. When a child member with a specific key has a different parent member than an already existing child member with the same key, the key will be reused but a new member will be generated. This equates to an invalid hierarchy.

Example 2

Date Hierarchies are common places to see invalid hierarchies. Naturally people tend to create the following hierarchy:

Year
Month
Week
Day

*** I am specifically talking about the "Year, MonthYear, WeekYear, and DayMonthYear" extract functions within DeepSee

As we know, a Week can be part of two months, or even two years. All of the other levels here (Year, Month, and Day) all fit in nicely to their parent and can't be part of two. When there is a hierarchy defined with a Week member in it like this, you can see unexpected results due to the way the DeepSee engine traverses the tree (Starting at Jan 3 2020, going up to Week 52 2019, which goes up to December 2019. Jan 3 2020 is not a child of December 2019, so the engine can remove these results).

The common solution is to create a new hierarchy with only the Week member in it. This will preserve the integrity of the original hierarchy but will also allow Week to be used within queries.

[DeepSeeButtons on Open Exchange](#) has a section in the generated report that will check these conditions and notify you if your hierarchies are invalid.

*** As of InterSystems IRIS 2020.3 and later, DeepSeeButtons is included in the product. Additional details can be found [in the documentation](#) ***

[#Analytics](#) [#Best Practices](#) [#Cubes](#) [#Data Model](#) [#Tips & Tricks](#) [#Caché](#) [#InterSystems IRIS](#) [#InterSystems IRIS BI \(DeepSee\)](#)

Source URL: <https://community.intersystems.com/post/designing-valid-hierarchies-deepsee>