

A Different Approach to Implementing Time Calculations in SSAS

[David Shroyer](#)
[OLAP Business Solutions](#)

April 2007 (Revised)

Applies to:

Microsoft SQL Server 2005 Analysis Services SP2

Summary: Analysis Services 2005 now includes built-in time intelligence enhancements which make it quick and easy to add time calculations to OLAP cubes. This paper will discuss the problems associated with the built-in enhancements and suggest a different approach for implementing global time calculations. Included with this paper is a .zip file called TimeCalculations.zip which includes the sample data mart and example SSAS project files.

Contents

[Introduction](#)
[The Business Problem](#)
[The Built-In Time Intelligence Wizard](#)
[Built-In Time Calculations Seen From a User Perspective](#)
[Advantages to Using Built-In Time Intelligence](#)
[Disadvantages to Using Built-In Time Intelligence](#)
[How Does the Time Calculation Dimension Work?](#)
[Creating a Shell Dimension](#)
[Adding the Time Calculation Dimension to the Cube](#)
[Adding the Time Calculations to the Cube](#)
[Verifying the Prior Year Calculation](#)
[Verifying the Year to Date Calculations](#)
[Adding a Different Type of User Defined Hierarchy](#)
[Adding the Attributes to the Dimension](#)
[Adding the Calculations](#)
[Conclusion](#)

Introduction

AS2K5 offers developers a built-in wizard which will automatically add global time calculations to SSAS cubes. These built-in MDX scripts are very limited in their calculation scope and prove to be cumbersome for end-users to use. Fortunately it is very easy to create your own time calculation dimension which can then be customized to the application's requirements.

The Business Problem

The fictitious company used in this example is called Sub Palace. Sub Palace is a small retail sandwich company headquartered in Northern California. The company has small retail outlets which have a limited sandwich and soda menu. The company has had some growth in the past few years and now has 10 retail outlets. The owner, Mr. Hanson, is trying to get a handle on his sales performance for each of his outlets.

A "development" SSAS cube was built which contains sales and profitability data for his ten stores. The cube has the following dimensions:

- Products
- Promotions
- Stores
- Time
- Measures

The requirements for the cube include providing users the ability to track the various cube measures by year over year performance and year to date performance.

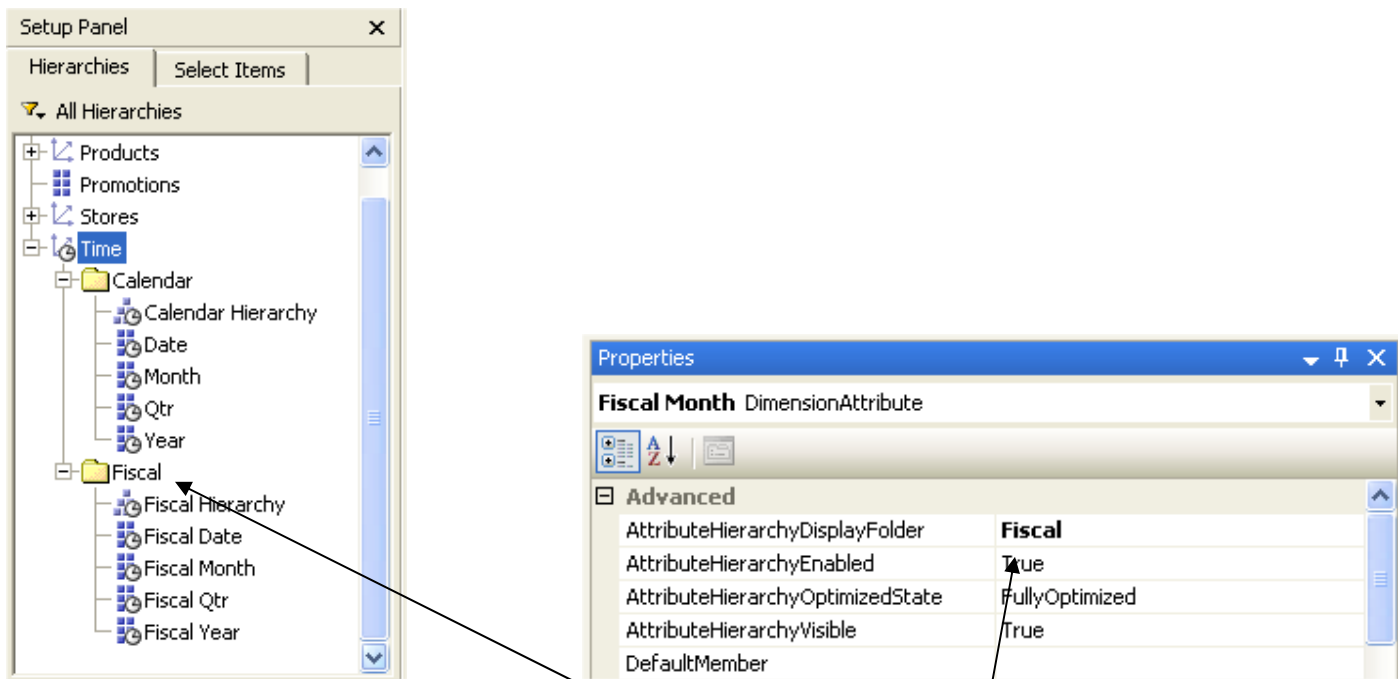
Included in the Timecalculations.zip file is a back up copy of the Sub Palace Data Mart. If you want to follow along with the examples in the paper, you will need to restore this SQL Server 2005 database. There are also three SSAS projects located in the .zip file.

- SubPalaceBase – This is the original “starting point” cube. If you want to follow along with the examples, this is the cube that you will want to use.
- SubPalaceWizard – This is the completed cube which uses the built-in time intelligence wizard. (First portion of the paper)
- SubPalaceManual – This is the completed cube which contains a manually created time dimension. (Solution suggested)

The Built-In Time Intelligence Wizard

The development SSAS database is set up with two user defined time hierarchies. There is a “calendar” time hierarchy and “fiscal” time hierarchy. Like most other business, SubPalace wants to track performance by both calendar and fiscal time periods.

The time dimension was structured so that the fiscal and calendar attribute dimensions are segmented into two separate folders. This makes it a little less confusing for users. They can be instructed to use either the “calendar hierarchy” or the “fiscal hierarchy”. This will insure that users do not accidentally use a fiscal attribute dimension in tandem with a calendar attribute dimension.



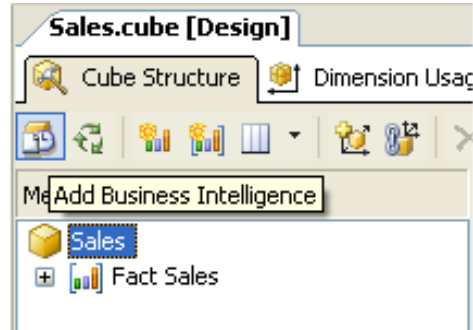
This screen shot illustrates how a user interacts with the dimensions in a front end tool such as ProClarity.

Note: To segment the attribute dimensions and hierarchies into folders, assign a value to the “AttributeHierarchyDisplayFolder” property of the attribute dimension. The key attribute dimension has been hidden to the user in the above example.

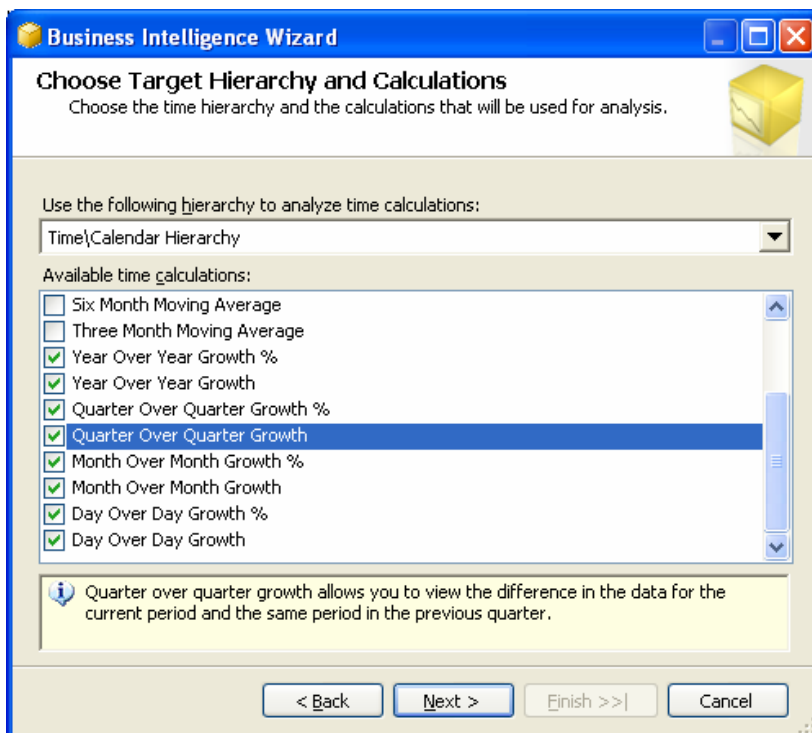
Adding Time Intelligence Using the Built-In Wizard

We are now going to use the “Add Business Intelligence” wizard to add some time intelligence to the cube. (A copy of this completed project is located in the accompanying zip file. The project name is SubPalaceWizard.) In this example we are interested in year over year calculations and year to date calculations.

- Double click on the Sales Cube to open the cube designer.
- Click the “Add Business Intelligence” button.

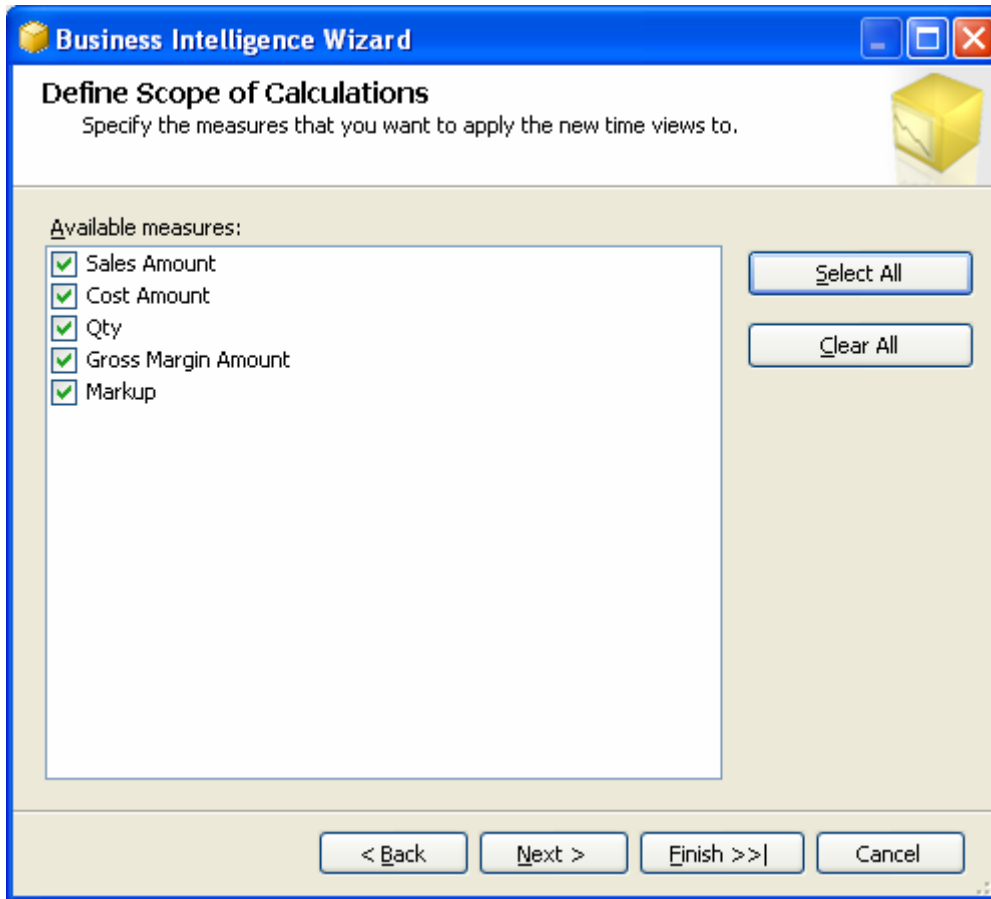


- Click past the Wizard Welcome screen.
- For “Cube Enhancement” choose the first option which should be “Define Time Intelligence”.
- In the top drop down list choose:
 - Year to Date
 - Year Over Year Growth %
 - Year Over Year Growth
 - Quarter Over Quarter Growth %
 - Quarter Over Quarter Growth
 - Month Over Month Growth %
 - Month Over Month Growth
 - Day Over Day Growth %
 - Day Over Day Growth



- Click Next

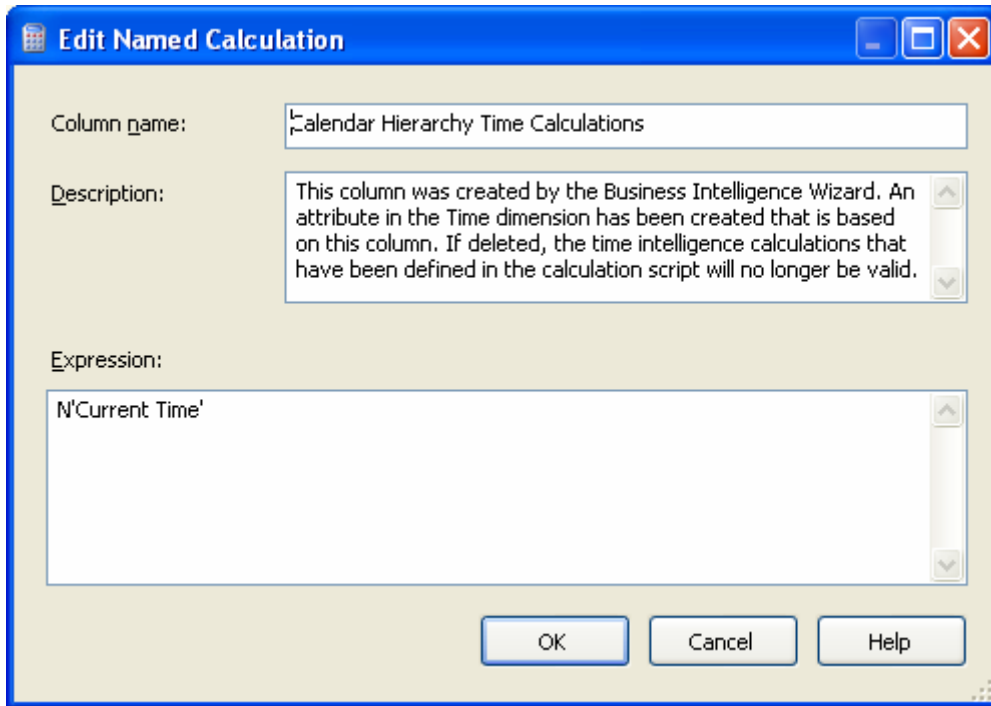
- On the “Define Scope Calculations” click on the “Select All” tab. (Because in this case we want the time calculations to apply to all of the measures.)



- Click Next and then Click Finish.

What Did the Wizard actually do?

- Added a new attribute dimension* - If you now open up the Time dimension you will notice a new attribute dimension called “Calendar Hierarchy Time Calculations”. This “shell” dimensions is what anchors all of the time calculations. If you click on the Browser tab and select this dimension in the drop down box you will notice that this new attribute dimension has one new member called “Current Time”.
- Added a field in the data source view (DSV)* – The new attribute dimension is using a new key that was added to the DimTime table in the DSV. If you open the DSV you will notice a new named calculation called “Calendar Hierarchy Time Calculations”. If you double click on the named calculation you will see the value as “Current Time”. This is the value that appears in the dimension as the base member.



3. *New calculation scripts were added* – New calculated members were added to the cube which performs the calculations that were selected in the wizard. Click on the Calculations tab. Click on the "Script View" icon and notice that there are now a bunch of MDX scripts which define the requested calculations.

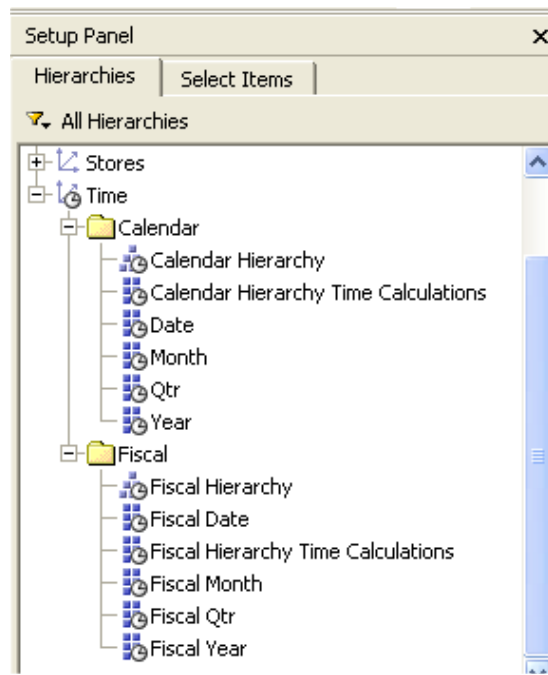
The first time we stepped through the time intelligence wizard we selected the "Calendar Hierarchy". We also want to provide the same calculations for the "Fiscal Hierarchy".

Step through the wizard one more time, but this time choose the "Fiscal Hierarchy". Notice that all of the same changes are made. There is new field created in the DSV, a new attribute dimension created in the time dimension and more MDX scripts added to the cube calculation script.

Save and Reprocess the cube.

Built In Time Calculations Seen From a User Perspective

Open your OLAP analysis tool (ProClarity is shown in the examples) and connect to the SubPalaceWizard database and Sales cube. You should now see the new "Time Calculations" dimensions within the Time dimension. Note that the users will have to be savvy enough to know that the "Calendar Hierarchy Time Calculations" only work with the "Calendar Hierarchy" and the "Fiscal Hierarchy Time Calculations" only work with the "Fiscal Hierarchy". This can be a little confusing for a lot of people. But fortunately the new time calculation dimensions naturally fell into their proper Fiscal and Calendar folders.



Now let's see how these calculations work....

The first calculation we will check is the "Year over Year Growth" calculations. For this example we put the "Calendar Hierarchy" dimension on the rows and the "Calendar Hierarchy Time Calculations" dimension on the columns.

	Current Time	Year Over Year Growth %	Year Over Year Growth
2002			
2003	\$3,592,058.11	1.#INF	\$3,592,058.11
Q1-2003	\$612,771.35	1.#INF	\$612,771.35
Q2-2003	\$757,344.49	1.#INF	\$757,344.49
Q3-2003	\$1,033,059.67	1.#INF	\$1,033,059.67
Q4-2003	\$1,188,882.60	1.#INF	\$1,188,882.60
2004	\$6,274,488.89	74.68%	\$2,682,430.78
Q1-2004	\$1,399,094.45	128.32%	\$786,323.10
Q2-2004	\$1,492,822.02	97.11%	\$735,477.53
Q3-2004	\$1,687,496.26	63.35%	\$654,436.59
Q4-2004	\$1,695,076.16	42.58%	\$506,193.56
2005	\$8,036,123.30	28.08%	\$1,761,634.41
2006	\$2,078,837.38	-74.13%	(\$5,957,285.92)

The calculations work except that the percent measure does not check for nulls before attempting to divide. (1.#INF)

Sales Amount (Calendar Hierarchy Time Calculations)					
	Quarter Over Quarter Growth %	Quarter Over Quarter Growth	Month Over Month Growth %	Month Over Month Growth	
2003	NA	NA	NA	NA	NA
Q1-2003	1.#INF	\$612,771.35	NA	NA	NA
Q2-2003	23.59%	\$144,573.14	NA	NA	NA
Q3-2003	36.41%	\$275,715.18	NA	NA	NA
Q4-2003	15.08%	\$155,822.93	NA	NA	NA
2004	NA	NA	NA	NA	NA
Q1-2004	17.68%	\$210,211.85	NA	NA	NA
January 2004	19.19%	\$68,356.81	0.58%	\$2,429.08	
February 2004	14.62%	\$60,000.98	10.77%	\$45,752.98	
March 2004	19.39%	\$81,854.06	7.16%	\$33,672.00	
Q2-2004	6.70%	\$93,727.57	NA	NA	NA
Q3-2004	13.04%	\$194,674.24	NA	NA	NA
Q4-2004	0.45%	\$7,579.90	NA	NA	NA
2005	NA	NA	NA	NA	NA

This next view shows the Quarter over Quarter and Month over Month calculations. You will notice that the Quarter and Month calculations are level dependant. They only provide values at the appropriate levels.

This can cause some real "usability" issues with users because they have to constantly be aware of what level in the time hierarchy they are displaying so they can pull over the appropriate time calculation. Not only is it confusing for users, but it also creates a "hard to read" grid because the calculations are located in different columns for different levels.

Advantages to Using Built-In Time Intelligence

- The template that "loads" the wizard is an xml file that can be modified. It is located at *c:\Program Files\Microsoft SQL Server\90\Tools\Templates\olap\1033\TimeIntelligence.xml*
- Allows developers to limit the measures that are included in the scope of the calculations. (Although now that the Aggregate function properly calculates ratios and distinct count measures, scoping is not as much of an issue as it was in AS2000).
- It is pretty easy.

Disadvantages to Using Built-In Time Intelligence

- Measures are "scoped" in the MDX. This means that any new measures or calculations that are added to cube which should be included in the time calculations will force a change in all of the time calculation formulas.
- The scope command is buried into each calculation rather than having one starting scope command that includes all of the time series calculations in a global statement.
- The calculations default to "NA" rather than NULL. This means that end user tools which suppress empty rows will not be able to suppress the empty rows because they are loaded with "NA" values. This also slows down the SSAS calculation engine.
- A new "time calculation" dimension has to be created for each user defined hierarchy. This also means a new attribute dimension and data source view calculation are also created for each user defined time hierarchy.
- Some very common and useful calculations are missing from the wizard, such as year to date variances.

How Does the Time Calculation Dimension Work?

Time Calculation dimensions use “shell dimensions” to perform the global time calculations across all measures. If you didn’t use a shell dimension for the time calculations and instead put them on the measures dimension, then a separate calculated member would have to be created for each measure.

Let’s say that we have a cube with 3 measures, Sales Amount, Qty, and Cost Amount. The requirements for the cube are that prior year calculations need to be provided for each of the measures. Using the “traditional” measures approach we would need to make 9 new calculated members on the measures dimension.

- Sales Amount Pr Yr
- Sales Amount Pr Yr Chg
- Sales Amount Pr Yr Chg %
- Qty Pr Yr
- Qty Pr Yr Chg
- Qty Pr Yr Chg %
- Cost Amount Pr Yr
- Cost Amount Pr Yr Chg
- Cost Amount Pr Yr Chg %

As you can imagine, the list of measures could grow enormously large. What if the requirements for the cube were to have 15 time calculations and there were 15 separate measures. Not only would it be a lot of work, to create 225 calculated members, but it wouldn’t be the easiest thing for users to navigate. (Can you imagine navigating a list of 225 measures?)

The use of a “shell dimension” solves this problem. A “shell dimension” is a dimension with one member that every record in the fact table is mapped against. Think of it as a single member that everything in the cube is “anchored” to. This allows the modification of the MDX so that rather than specifying a measure, we can instead specify this one member as the anchor.

Sales Amount Pr Yr Chg = ([Time].[Calendar Hierarchy].CurrentMember.Lag(1), [Measures].[Sales Amount]), (ParallelPeriod([Time].[Calendar Hierarchy].[Year], 1), [Measures].[Sales Amount]))

In the MDX script for “Sales Amount Pr Yr Chg”, the formula is saying “Go back one year for Sales Amount and grab the variance”. The “**what**” in the formula is [Measures].[Sales Amount]. If we create a time calculation shell dimension which has one member called “Current Period”, then we could use that in the “what” portion of the formula.

([Time].[Calendar Hierarchy].CurrentMember.Lag(1), [Time Calculations].[Current Period]), (ParallelPeriod([Time].[Calendar Hierarchy].[Year], 1), [Time Calculations].[Current Period]))

This formula now is saying “Go back one year from the **current period**”. Because we are no longer specifying a particular measure, the calculation will calculate for all measures. The trick here is to change the “anchor” of the formula from a measure to the base member of the “shell dimension”. This new calculation also has to reside on the shell dimension and not on the measures dimension.

Creating a Shell Dimension

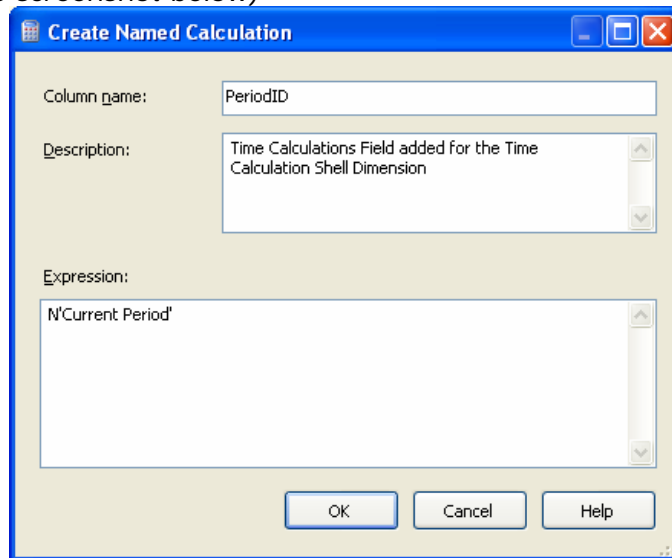
The first step to manually adding time calculations to a SSAS cube is to create the shell dimension which will hold the calculations. (Note: this completed project is called SubPalaceManual) The time intelligence wizard places this shell dimension within the time dimension. This can be confusing to users for the following reason. Oftentimes the time dimension will have multiple user defined hierarchies. Usually the “rule” for using these hierarchies is that the user should only use one hierarchy at a time. Otherwise they will be crossing fiscal dates with calendar dates. If the time calculation shell dimension is located within the time dimension then this rule is broken. The new rule would have to be “you can only use one time hierarchy at a time, but you can always use the one called time calculations.” I find it much easier to create a physically separate time calculation shell dimension. This provides for greater usability from a

user standpoint. (And it doesn't affect cube performance.) Even though we are going to create a physically separate shell dimension, we are going to build the new dimension from our time dimension table. (Like the wizard)

Modifying the Time Dimension Table

In this example, we are going to add a named calculation to the DimTime table which will be the source for our new Time Calculations dimension.

- Open the Data Source View
- Right click on the DimTime table and choose "Named Calculation"
- Name the column "PeriodID"
- The Expression equals N'Current Period'
- This places a column in the DimTime table where each record contains "Current Period" as the value for the PeriodID
- Click OK (see screenshot below)

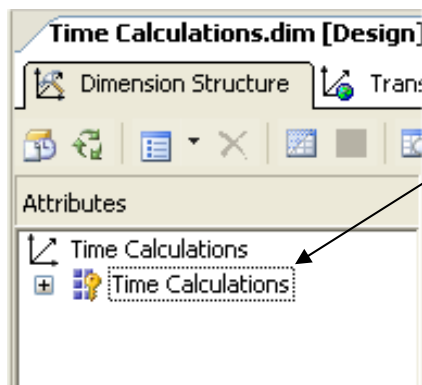


Creating the TimeCalculation Dimension

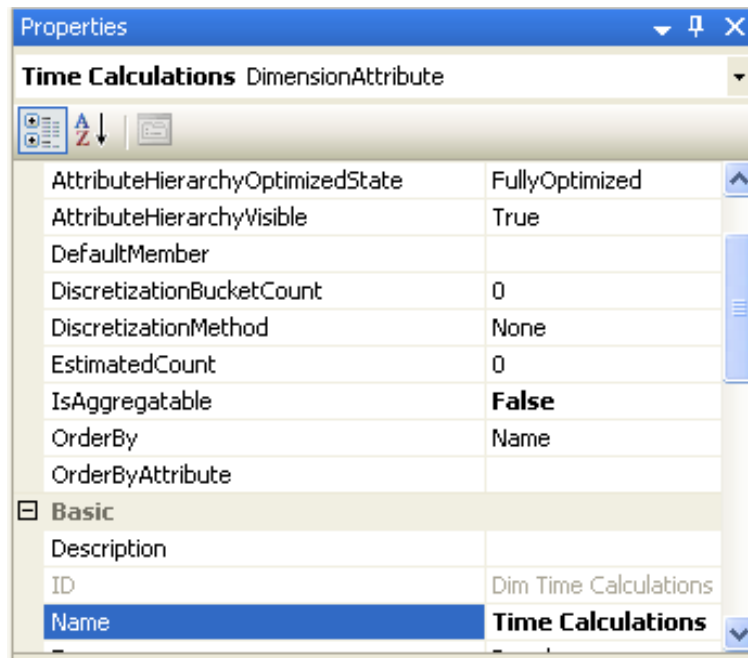
- Right click on the Dimension folder, within the Solution Explorer, and choose "New Dimension".
- Skip past the welcome screen by choosing Next
- Uncheck the "Auto Build" option and click Next
- Select the Data Source View SubPalace
- Select "Standard Dimension" when prompted for a Dimension Type. Click Next
- The "Select the Main Dimension Table" screen should have the DimTime selected as the main table. The key column is the PeriodId field and the member name column should be the PeriodID field. (As shown below)



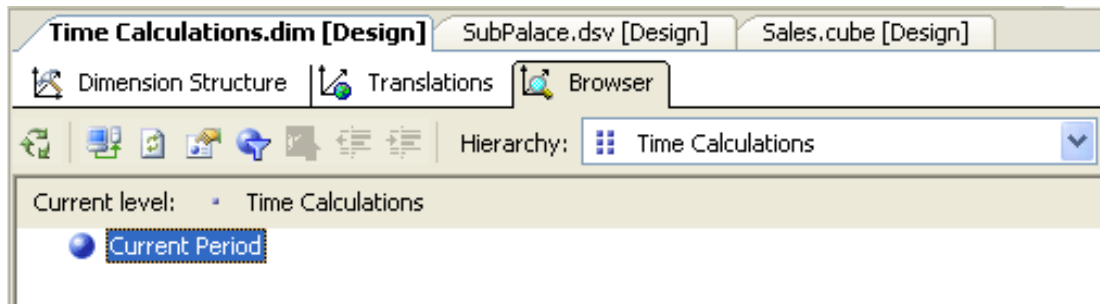
- Click Next and then Next again to get to the last screen
- Name the Dimension **"Time Calculations"** and click OK
- Rename the "Dim Time Calculations" attribute dimension to "Time Calculations"



- Because this dimension is going to be used exclusively for calculations, we do not want to have an "All" level. It wouldn't make sense to have a total for current period + prior year + ytd, etc.
- Click on the Time Calculations Attribute dimension and set the property IsAggregatable equal to False. (Shown below) This will not create an "All Level".



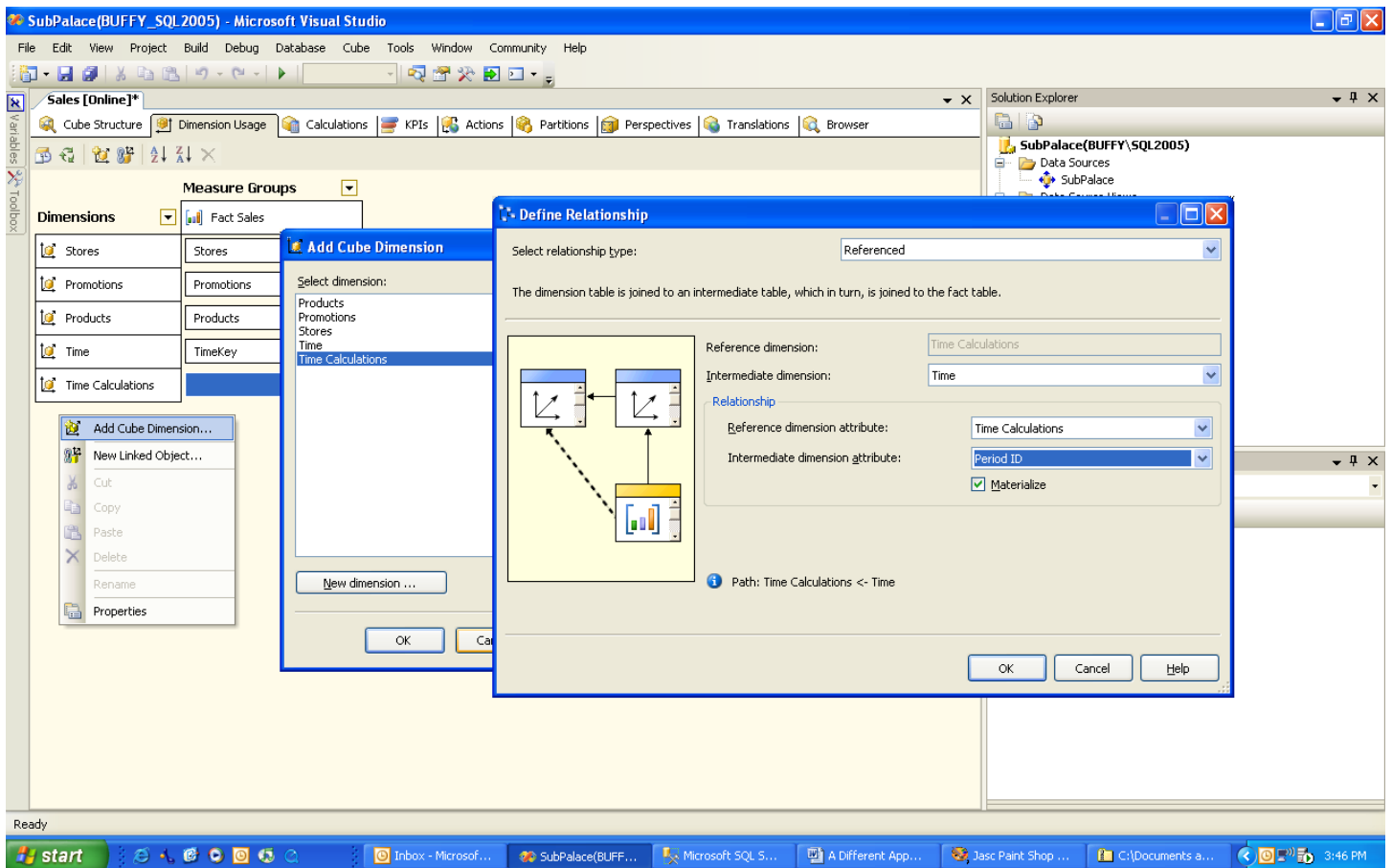
- Save and Process the dimension.
- When you browse the dimension you should see one member called “Current Period” and no total at the top.



Adding the Time Calculation Dimension to the Cube

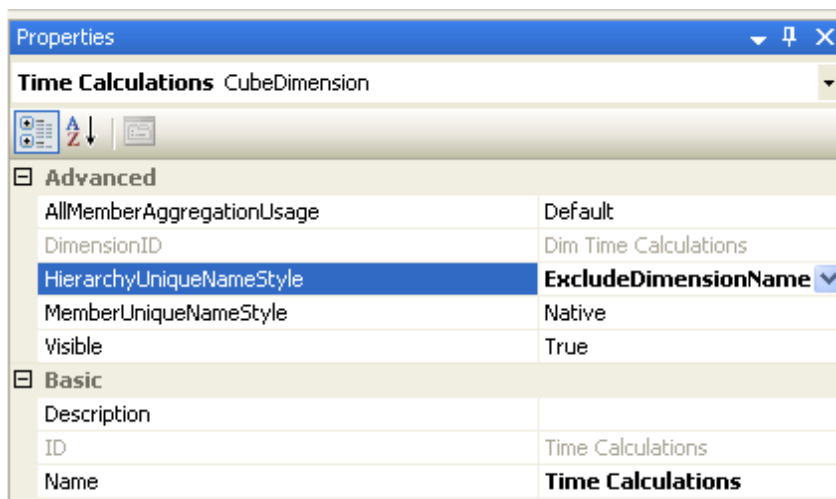
The next step is to add the time calculation dimension to the sales cube. Because the PeriodID key is not located in the fact table, we are going to use a “Referenced” type relationship when we add the dimension to the cube.

- Double click on the sales.cube in the solution explorer to open the sales cube.
- Click on the Dimension Usage tab.
- Right click in an open area and select “Add Cube Dimension”.
- Select “Time Calculations” and click OK.
- Click on the Ellipse and insure that the relationship between the dimensions and the Fact Sales cube is based on the PeriodID.
- The Relationship Type should be “Referenced”.
- The Intermediate dimension is “Time”.
- The Referenced dimension attribute is “Time Calculations” and the Intermediate dimension attribute is “PeriodID”. (Shown below)



Because this new time calculation dimension does not have any user defined hierarchies, the MDX will be simplified if we do not include the dimension name when referring to the dimension in MDX.

- Click on the Cube Structure tab.
- Highlight the Time Calculation Hierarchy in the lower left hand corner of the screen.
- Set the HierarchyUniqueNameStyle property to "ExcludeDimensionName".

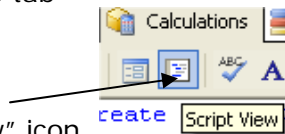


- Save and Process the Cube

Adding the Time Calculations to the Cube

The first step in adding the time calculations is to create the members. Once the members are created, we can then use a script to “populate” the empty calculated measure.

- Open the Cube Editor
- Click on the Calculations tab



- Click on the “Script View” icon.
- Use the MDX below to create the calculations.
- These CREATE statements will create the calculations.

```
-- Time Series Calculations using the SCOPE command to fix on a sub cube
-- Create the calculated members
```

```
CREATE MEMBER CURRENTCUBE.[Time Calculations].[Prior Year] AS Null;
CREATE MEMBER CURRENTCUBE.[Time Calculations].[Prior Year Chg] AS Null;
CREATE MEMBER CURRENTCUBE.[Time Calculations].[Prior Year Chg %] AS
IIF([Time Calculations].[Prior Year]=0,NULL,
[Time Calculations].[Prior Year Chg] / [Time Calculations].[Prior Year]),
FORMAT_STRING="Percent";
CREATE MEMBER CURRENTCUBE.[Time Calculations].[Prior Period] AS Null;
CREATE MEMBER CURRENTCUBE.[Time Calculations].[Prior Period Chg] AS Null;
CREATE MEMBER CURRENTCUBE.[Time Calculations].[Prior Period Chg %] AS
IIF([Time Calculations].[Prior Period]=0,NULL,
[Time Calculations].[Prior Period Chg] / [Time Calculations].[Prior Period]),
FORMAT_STRING="Percent";
CREATE MEMBER CURRENTCUBE.[Time Calculations].[YTD] AS Null;
CREATE MEMBER CURRENTCUBE.[Time Calculations].[YTD Pr Yr] AS Null;
CREATE MEMBER CURRENTCUBE.[Time Calculations].[YTD Pr Yr Chg] AS Null;
CREATE MEMBER CURRENTCUBE.[Time Calculations].[YTD Pr Yr Chg %] AS
IIF([Time Calculations].[YTD Pr Yr]=0,NULL,
[Time Calculations].[YTD Pr Yr Chg] / [Time Calculations].[YTD Pr Yr]),
FORMAT_STRING="Percent";
```

The first thing to notice is that this calculation is being stored on the Time Calculation dimension, not the Measures dimension.

```
CREATE MEMBER CURRENTCUBE.[Time Calculations].[Prior Year]
```

The next step is to create the script which actually performs the time calculations. Add the MDX below to the calculation script AFTER the above create commands. The first Scope command allows the time calculations to only apply to a subset of the measures. Because the other Scope commands are nested within this one command, this approach allows for ease of maintenance because the logic is only listed in one location.

The following script will limit the time calculations so they do not include the “Markup” calculated measure. The Scope function does not include the measure “Markup”, therefore the time calculations will return NULL when “Markup” is selected.

```
-- Scope for the specific Measures
-- You can use the commented out approach if you want to specify the exact measures you
-- want included in the calculations.
--Scope({[Measures].[Sales Amount],
-- [Measures].[Cost Amount],
-- [Measures].[Qty],
-- [Measures].[Gross Margin Amount]});
--If you use this scope statement you are saying that you want ALL of the measures
```

```
-- from the Fact Sales measure group EXCEPT the Markup Measure
-- If in the future you add a new measure to the measure group, it will
-- automatically be included
```

```
Scope(MeasureGroupMeasures("Fact Sales") - {[Measures].[Markup]});
```

```
-- Calendar Time Hierarchy
-- This piece scopes on the calendar hierarchy by scoping on the highest level (Year) and
the key
-- Because the calendar hierarchy attributes are related, this includes all members of
that user defined hierarchy
-- This is faster than scoping on the actual user defined hierarchy
-- This approach only works if non of the attribute dimensions cross (or shared) between
hierarchies.
```

```
Scope([Time].[Calendar Year].[Calendar Year].members,[Time].[TimeKey].members);
```

```
-- PRIOR YEAR CALCULATIONS
```

```
([Time Calculations].[Prior Year]=
    (ParallelPeriod([Time].[Calendar Hierarchy].[Year], 1,
        [Time].[Calendar Hierarchy].currentmember)
        ,[Time Calculations].[Current Period])
);
```

```
-- PRIOR PERIOD CALCULATIONS
```

```
([Time Calculations].[Prior Period]=
    IIF(([Time].[Calendar Hierarchy].currentmember.lag(1),[Time
Calculations].[Current Period])=0,null,
    ([Time].[Calendar Hierarchy].currentmember.lag(1),
    [Time Calculations].[Current Period]))
);
```

```
-- YTD CALCULATIONS
```

```
([Time Calculations].[YTD]=
    Aggregate(
        CrossJoin({[Time Calculations].[Current Period]},
            PeriodsToDate([Time].[Calendar Hierarchy].[Year],
                [Time].[Calendar Hierarchy].CurrentMember))
    )
);
```

```
-- YTD PRIOR YEAR
```

```
([Time Calculations].[YTD Pr Yr]=
    Aggregate(
        Crossjoin({[Time Calculations].[Current Period]},
            PeriodsToDate(
                [Time].[Calendar Hierarchy].[Year],
                ParallelPeriod(
                    [Time].[Calendar Hierarchy].[Year],1,
                    [Time].[Calendar Hierarchy].CurrentMember))
            )
    )
);
```

```
End Scope;
```

```
-- Fiscal Time Hierarchy
```

```
Scope([Time].[Fiscal Year].[Fiscal Year].members, [Time].[TimeKey].members);
```

```
-- PRIOR YEAR CALCULATIONS
```

```
([Time Calculations].[Prior Year]=
    (ParallelPeriod([Time].[Fiscal Hierarchy].[Fiscal Year], 1,
        [Time].[Fiscal Hierarchy].CurrentMember)
        ,[Time Calculations].[Current Period])
);
```

```
-- PRIOR PERIOD CALCULATIONS
```

```
([Time Calculations].[Prior Period]=
    IIF(([Time].[Fiscal Hierarchy].currentmember.lag(1),
```

```

        [Time Calculations].[Current Period]) =0,null,
        ([Time].[Fiscal Hierarchy].currentmember.lag(1),
        [Time Calculations].[Current Period]))
    );
-- YTD CALCULATIONS
([Time Calculations].[YTD]=
    Aggregate(
        CrossJoin({[Time Calculations].&[Current Period]},
            PeriodsToDate([Time].[Fiscal Hierarchy].[Fiscal Year],
                [Time].[Fiscal Hierarchy].CurrentMember))
        )
    );
-- YTD PRIOR YEAR CALCULATIONS
([Time Calculations].[YTD Pr Yr]=
    Aggregate(
        CrossJoin({[Time Calculations].[Current Period]},
            PeriodsToDate(
                [Time].[Fiscal Hierarchy].[Fiscal Year],
                ParallelPeriod(
                    [Time].[Fiscal Hierarchy].[Fiscal Year],1,
                    [Time].[Fiscal Hierarchy].CurrentMember)))
        )
    );
End Scope;

```

-- Now do the Variances which do not depend on a particular hierarchy

-- Prior Year Chg

```

([Time Calculations].[Prior Year Chg]=
[Time Calculations].[Current Period] - [Time Calculations].[Prior Year]);

```

-- Prior Period Chg

```

([Time Calculations].[Prior Period Chg] =
[Time Calculations].[Current Period] - [Time Calculations].[Prior Period]);

```

-- YTD Prior Year Chg

```

([Time Calculations].[YTD Pr Yr Chg] =
[Time Calculations].[YTD] - [Time Calculations].[YTD Pr Yr]);

```

End scope;

This approach assumes that users will NOT be using both user defined hierarchies at the same time.

The "rule" for users of the cube is that they can only use one user defined time hierarchy at a time. But what this approach allows is one set of calculations that apply to both user defined hierarchies.

- Save and Deploy the Cube

Verifying the Prior Year Calculation

- Click on the Browser window or open your favorite client side tool.
- Place the Time Calculations dimension on the columns and choose the "Current Period" and "Prior Year" members.
- Place the "Calendar Hierarchy" dimension on the rows and drill down on "2004" and "2005".

Sales Amount (Time Calculations)				
	Current Period	Prior Year	Prior Year Chg	Prior Year Chg %
All Years	\$19,981,507.68		\$19,981,507.68	
2002				
2003	\$3,592,058.11		\$3,592,058.11	
2004	\$6,274,488.89	\$3,592,058.11	\$2,682,430.78	74.68%
Q1-2004	\$1,399,094.45	\$612,771.35	\$786,323.10	128.32%
Q2-2004	\$1,492,822.02	\$757,344.49	\$735,477.53	97.11%
Q3-2004	\$1,687,496.26	\$1,033,059.67	\$654,436.59	63.35%
Q4-2004	\$1,695,076.16	\$1,188,882.60	\$506,193.56	42.58%
2005	\$8,036,123.30	\$6,274,488.89	\$1,761,634.41	28.08%
Q1-2005	\$1,819,744.54	\$1,399,094.45	\$420,650.09	30.07%
Q2-2005	\$1,905,177.71	\$1,492,822.02	\$412,355.69	27.62%
Q3-2005	\$2,170,768.98	\$1,687,496.26	\$483,272.72	28.64%
Q4-2005	\$2,140,432.07	\$1,695,076.16	\$445,355.91	26.27%
2006	\$2,078,837.38	\$8,036,123.30	(\$5,957,285.92)	-74.13%

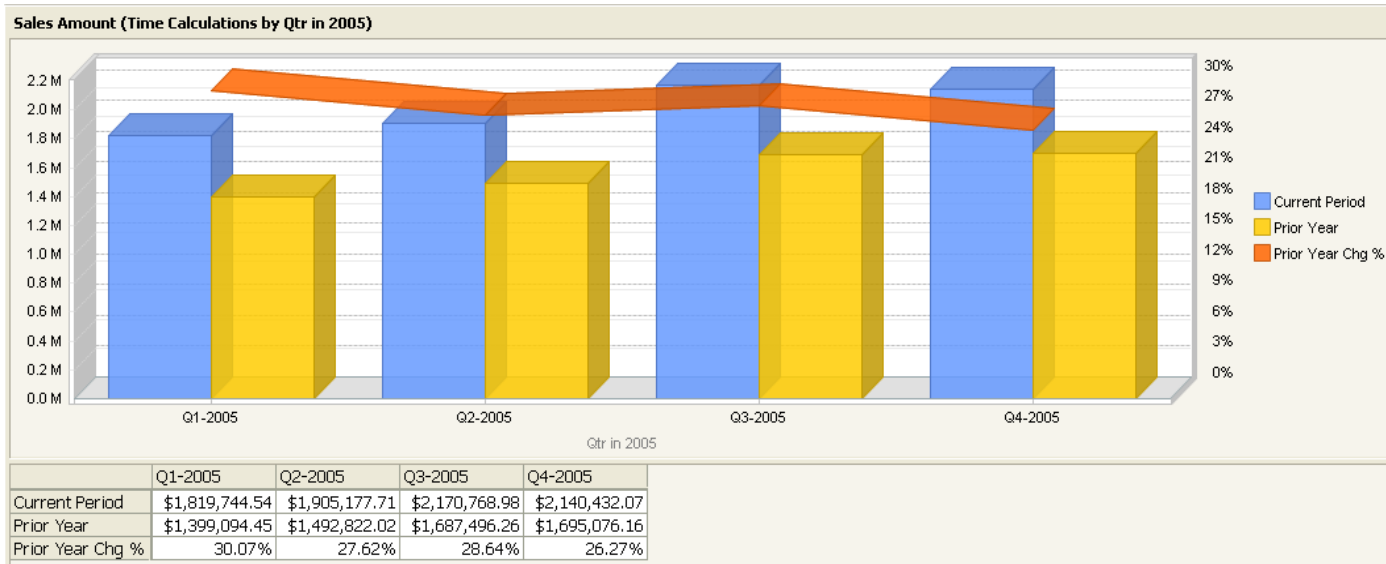
- The "Prior Year" calculation is working at both the year and quarter level of the calendar hierarchy. In fact, if you drill down you will notice that it works at all levels within the cube.
- Now place the "Fiscal Hierarchy" on the Rows and move the "Calendar Hierarchy" to the background.
- Remember to select from the ALL level on the "Calendar Hierarchy" since the rule is that we can only use one at a time.

Sales Amount (Time Calculations)		
	Current Period	Prior Year
All Fiscal Years	\$19,981,507.68	
FY2003	\$1,370,115.84	
Q1-2003		
Q2-2003		
Q3-2003	\$612,771.35	
Q4-2003	\$757,344.49	
FY2004	\$5,113,858.74	\$1,370,115.84
Q1-2004	\$1,033,059.67	
Q2-2004	\$1,188,882.60	
Q3-2004	\$1,399,094.45	\$612,771.35
Q4-2004	\$1,492,822.02	\$757,344.49
FY2005	\$7,107,494.67	\$5,113,858.74
Q1-2005	\$1,687,496.26	\$1,033,059.67
Q2-2005	\$1,695,076.16	\$1,188,882.60
Q3-2005	\$1,819,744.54	\$1,399,094.45
Q4-2005	\$1,905,177.71	\$1,492,822.02
FY2006	\$6,390,038.43	\$7,107,494.67
FY2007		\$6,390,038.43

- The screen shot above shows that the "Prior Year" calculation also calculates properly when the "Fiscal Hierarchy" dimension is displayed.

A calculation that was added in the manual creation of the time calculation dimension that was not offered in the canned time intelligence wizard was the "Prior Year" calculation. Variances between the "Current Year" and "Prior Year" are important, but it is also valuable to display the "Prior Year".

The graph below shows a ProClarity view which has the "Time Calculation" dimension on the rows and the "Calendar Hierarchy" on the columns. The "Prior Year" calculation allows for the creation of powerful year over year visual comparisons. Notice how in this simple view, the user can see the "Current Period", the "Prior Year" period, and the percent change all in one view.



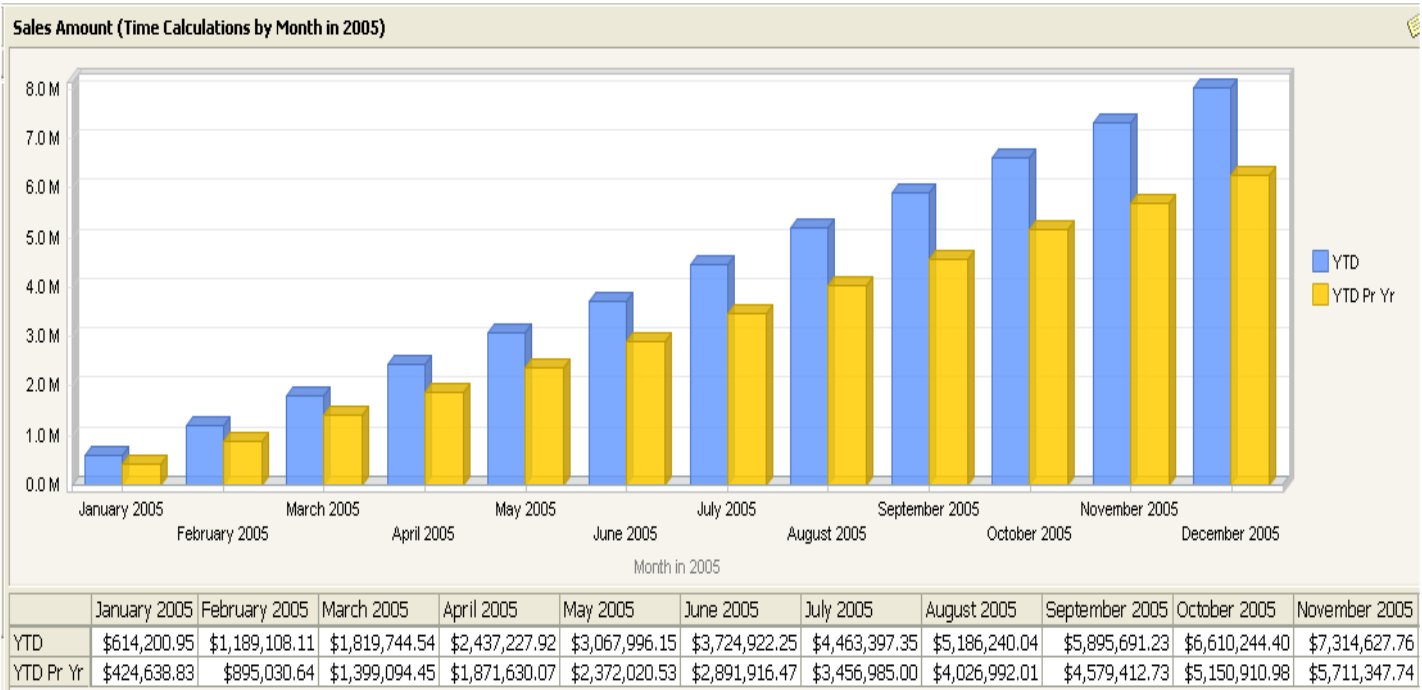
Verifying the YTD Calculations

- Click on the Browser window or open your favorite client side tool
- Place the "Time Calculations" dimension on the columns and choose "Current Period", "YTD", "YTD Pr Yr", "YTD Pr Yr Chg", and "YTD Pr Yr Chg %".
- Place the "Calendar Hierarchy" dimension on the rows and drill down on "2004" and "2005".

Sales Amount (Time Calculations)

	Current Period	YTD	YTD Pr Yr	YTD Pr Yr Chg	YTD Pr Yr Chg %
2005	\$8,036,123.30	\$8,036,123.30	\$6,274,488.89	\$1,761,634.41	28.08%
Q1-2005	\$1,819,744.54	\$1,819,744.54	\$1,399,094.45	\$420,650.09	30.07%
January 2005	\$614,200.95	\$614,200.95	\$424,638.83	\$189,562.12	44.64%
February 2005	\$574,907.16	\$1,189,108.11	\$895,030.64	\$294,077.47	32.86%
March 2005	\$630,636.43	\$1,819,744.54	\$1,399,094.45	\$420,650.09	30.07%
Q2-2005	\$1,905,177.71	\$3,724,922.25	\$2,891,916.47	\$833,005.78	28.80%
April 2005	\$617,483.38	\$2,437,227.92	\$1,871,630.07	\$565,597.85	30.22%
May 2005	\$630,768.23	\$3,067,996.15	\$2,372,020.53	\$695,975.62	29.34%
June 2005	\$656,926.10	\$3,724,922.25	\$2,891,916.47	\$833,005.78	28.80%
Q3-2005	\$2,170,768.98	\$5,895,691.23	\$4,579,412.73	\$1,316,278.50	28.74%
Q4-2005	\$2,140,432.07	\$8,036,123.30	\$6,274,488.89	\$1,761,634.41	28.08%

The screenshot above validates that the YTD calculations are working properly. The "YTD Pr Yr" calculation is another very powerful calculation because it allows analysts to create charts like the one shown below.



Adding a Different Type of User Defined Hierarchy

So far this solution has addressed the use of user defined hierarchies which contain the year within the hierarchy. These are the most common type of time hierarchies. But many users will tell you that they really want to use the “attribute dimensions” individually to perform their analysis. This is almost always because they want to put the years on one axis and the months or weeks on another. As you can imagine, this causes a lot more complexity when writing global time calculations. Now, rather than having just one hierarchy to “worry about”, you might have 3 or 4 independent attributes which can have numerous “intersections”.

Since the majority of users want time attribute dimensions so they can pivot the year attribute dimension against the others, I’ve found that the easiest way to model this behavior is to create a “time periods” user defined hierarchy which contains a hierarchy that does not contain the year.

- Fiscal Qtr – such as Q1
- Fiscal Month – such as July
- Fiscal Week – such as Week 1

Another separate attribute dimension called “Fiscal Years” could be created which works in tandem with the user defined “time periods” hierarchy. This allows users to put the hierarchy on the rows, and thus have the ability to drill down, while the years are put on the columns. These two items can be put into a folder so users can easily see that they “work together”.

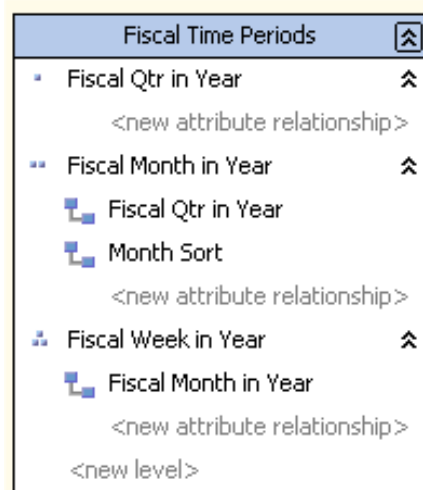
The grid below shows an example grid with the “fiscal time periods” hierarchy on the rows and the “fiscal years” attribute dimension on the columns. Also nested on the columns is the Time Calculations dimension.

Sales Amount								
	FY2005				FY2006			
	Current Period	Prior Year	Prior Year Chg	Prior Year Chg %	Current Period	Prior Year	Prior Year Chg	Prior Year Chg %
Year Total	\$7,066,417.46	\$5,069,760.74	\$1,996,656.72	39.38%	\$6,487,161.66	\$7,066,417.46	(\$579,255.80)	-8.20%
Q1	\$1,669,580.27	\$1,020,240.22	\$649,340.05	63.65%	\$2,141,182.75	\$1,669,580.27	\$471,602.48	28.25%
Jul	\$640,176.45	\$382,165.53	\$258,010.92	67.51%	\$835,598.33	\$640,176.45	\$195,421.88	30.53%
Week 1	\$130,747.41	\$74,959.31	\$55,788.10	74.42%	\$172,428.41	\$130,747.41	\$41,681.00	31.88%
Week 2	\$131,070.20	\$76,849.97	\$54,220.23	70.55%	\$164,744.00	\$131,070.20	\$33,673.80	25.69%
Week 3	\$121,799.93	\$76,276.36	\$45,523.57	59.68%	\$161,239.27	\$121,799.93	\$39,439.34	32.38%
Week 4	\$130,733.64	\$77,884.16	\$52,849.48	67.86%	\$165,065.14	\$130,733.64	\$34,331.50	26.26%
Week 5	\$125,825.27	\$76,195.73	\$49,629.54	65.13%	\$172,121.51	\$125,825.27	\$46,296.24	36.79%
Aug	\$382,934.77	\$239,090.60	\$143,844.17	60.16%	\$490,283.58	\$382,934.77	\$107,348.81	28.03%
Week 6	\$130,641.10	\$76,486.03	\$54,155.07	70.80%	\$165,962.43	\$130,641.10	\$35,321.33	27.04%
Week 7	\$125,710.69	\$80,769.35	\$44,941.34	55.64%	\$159,383.50	\$125,710.69	\$33,672.81	26.79%
Week 8	\$126,582.98	\$81,835.22	\$44,747.76	54.68%	\$164,937.65	\$126,582.98	\$38,354.67	30.30%
Sep	\$646,469.05	\$398,984.09	\$247,484.96	62.03%	\$815,300.84	\$646,469.05	\$168,831.79	26.12%
Q2	\$1,684,187.90	\$1,174,904.42	\$509,283.48	43.35%	\$2,130,692.03	\$1,684,187.90	\$446,504.13	26.51%
Oct	\$518,242.75	\$320,553.30	\$197,689.45	61.67%	\$645,919.93	\$518,242.75	\$127,677.18	24.64%
Dec	\$516,539.90	\$383,464.10	\$133,075.80	34.70%	\$655,368.38	\$516,539.90	\$138,828.48	26.88%
Nov	\$649,405.25	\$470,887.02	\$178,518.23	37.91%	\$829,403.72	\$649,405.25	\$179,998.47	27.72%
Q3	\$1,799,325.38	\$1,388,425.82	\$410,899.56	29.59%	\$2,099,819.47	\$1,799,325.38	\$300,494.09	16.70%
Q4	\$1,913,323.91	\$1,486,190.28	\$427,133.63	28.74%	\$115,467.41	\$1,913,323.91	(\$1,797,856.50)	-93.97%

Adding the Attributes to the Dimension

The first step to create the “fiscal time period” hierarchy is to first add the attributes to the time dimension.

1. Drag the FQtr, FMonth, and FWeek fields over to the left attributes pane to create the attributes.
2. Name FQtr as “Fiscal Qtr in Year”, FMonth as “Fiscal Month in Year”, and Fweek as “Fiscal Week in Year”.
3. Create the user defined hierarchy with the following levels and relationships.



4. Set the “allmembername” of the hierarchy to “Year Total”
5. Set the “Fiscal Month in Year” attribute to Sort using the “Month Sort” attribute.
6. Hide the fiscal qtr in year, fiscal month in year, and fiscal week in year attributes.
7. Put the Fiscal Time Periods hierarchy and the Fiscal Years attribute in a folder called “Fiscal Attributes”
8. Save and Process the dimension.

Adding the Calculations

The next step is to add the calculations to the cube. Go to the “calculations” tab and paste the following code ABOVE the “the variance” section. (And below the fiscal and calendar hierarchy calcs)

```

-- Fiscal Time Periods Hierarchy
Scope([Time].[Fiscal Years].[Fiscal Years].members, [Time].[TimeKey].members);

-- YTD Calculation
([Time Calculations].[YTD]=
    Aggregate(
        CrossJoin({[Time Calculations].[Current Period]},
            {NULL:[Time].[Fiscal Time Periods].CurrentMember})
    )
);

-- PRIOR PERIOD CALCULATIONS
([Time Calculations].[Prior Period]=
    IIF(([Time].[Fiscal Time Periods].currentmember.lag(1),
        [Time Calculations].[Current Period])=0,null,
        ([Time].[Fiscal Time Periods].currentmember.lag(1),
        [Time Calculations].[Current Period]))
);

-- Scope for Qtr1
Scope([Time].[Fiscal Qtr in Year].[Q1]);
([Time Calculations].[Prior Period]=
    ([Time].[Fiscal Qtr in Year].[Q4],
    [Time].[Fiscal Years].prevmember,[Time Calculations].[Current Period]));
End Scope;

-- Scope for JUL
Scope([Time].[Fiscal Month in Year].[Jul]);
([Time Calculations].[Prior Period]=
    ([Time].[Fiscal Month in Year].[Jun],
    [Time].[Fiscal Years].prevmember,[Time Calculations].[Current Period]));
End Scope;

-- PRIOR YEAR CALCULATIONS
([Time Calculations].[Prior Year]=
    ([Time].[Fiscal Years].prevmember
    ,[Time Calculations].[Current Period])
);

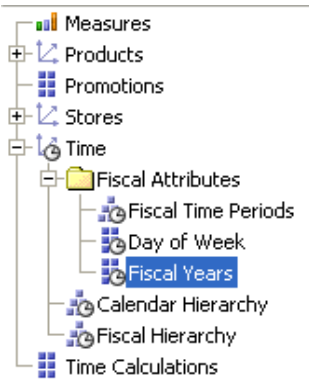
-- YTD LY
([Time Calculations].[YTD Pr Yr]=
    Aggregate(
        CrossJoin({[Time Calculations].[Current Period]},
            {NULL:[Time].[Fiscal Time Periods].CurrentMember},
            {[Time].[Fiscal Years].prevmember})
    )
);
End Scope;

```

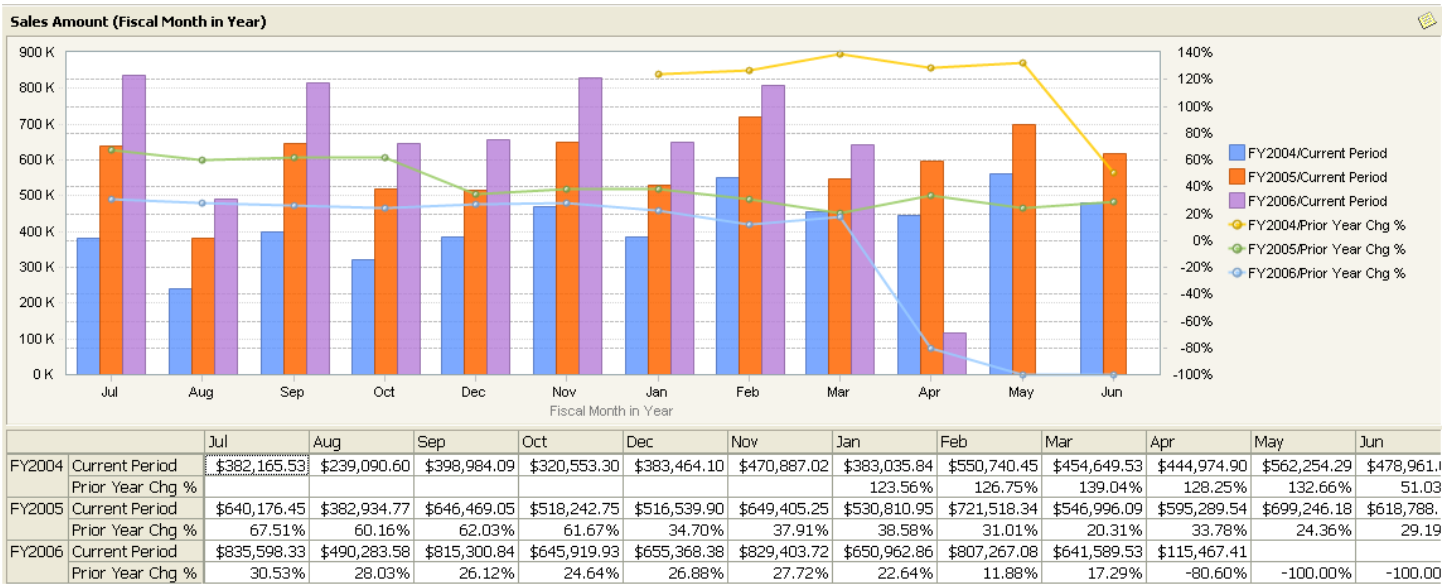
After saving and deploying the project you should be able to see the new folder under the Time dimension. Within the folder you should see the Fiscal Time Periods hierarchy and the Fiscal Years attribute.

Now users can use the following combinations with the Time Calculations dimension:

- Calendar Hierarchy with Time Calculations Dimension
- Fiscal Hierarchy with the Time Calculations Dimension
- Fiscal Time Periods Hierarchy AND Fiscal Years Attribute with the Time Calculations Dimension.
- Day of Week Attribute with any of the above.



The graph and grid below shows the type of analysis than can now be accomplished with the new Fiscal Time Periods hierarchy.



Users can also use the "Day of Week" attribute with the "Fiscal Time Periods" hierarchy to see data at the day level. This is accomplished by nesting the "Fiscal Time Periods" and the "Day of Week" dimensions on the rows.

Sales Amount

		FY2005				FY2006			
		Current Period	Prior Year	Prior Year Chg	Prior Year Chg %	Current Period	Prior Year	Prior Year Chg	Prior Year Chg %
Jul	Sunday	\$73,432.06	\$42,591.41	\$30,840.65	72.41%	\$92,232.44	\$73,432.06	\$18,800.38	25.60%
	Monday	\$75,833.64	\$44,353.72	\$31,479.92	70.97%	\$90,640.17	\$75,833.64	\$14,806.53	19.53%
	Tuesday	\$70,774.24	\$44,002.69	\$26,771.55	60.84%	\$91,671.88	\$70,774.24	\$20,897.64	29.53%
	Wednesday	\$74,951.87	\$46,875.06	\$28,076.81	59.90%	\$95,771.09	\$74,951.87	\$20,819.22	27.78%
	Thursday	\$74,983.78	\$44,096.18	\$30,887.60	70.05%	\$102,115.74	\$74,983.78	\$27,131.96	36.18%
	Friday	\$72,414.87	\$44,865.56	\$27,549.31	61.40%	\$93,435.46	\$72,414.87	\$21,020.59	29.03%
	Saturday	\$71,960.72	\$39,185.18	\$32,775.54	83.64%	\$97,610.04	\$71,960.72	\$25,649.32	35.64%
Week 1	Sunday	\$16,980.33	\$9,905.89	\$7,074.44	71.42%	\$24,009.99	\$16,980.33	\$7,029.66	41.40%
	Monday	\$19,644.40	\$11,948.02	\$7,696.38	64.42%	\$25,396.46	\$19,644.40	\$5,752.06	29.28%
	Tuesday	\$18,808.49	\$11,257.38	\$7,551.11	67.08%	\$20,531.47	\$18,808.49	\$1,722.98	9.16%
	Wednesday	\$17,593.13	\$11,209.93	\$6,383.20	56.94%	\$25,390.07	\$17,593.13	\$7,796.94	44.32%
	Thursday	\$19,002.03	\$11,045.32	\$7,956.71	72.04%	\$25,805.23	\$19,002.03	\$6,803.20	35.80%
	Friday	\$19,848.87	\$9,970.66	\$9,878.21	99.07%	\$24,598.48	\$19,848.87	\$4,749.61	23.93%
	Saturday	\$18,870.16	\$9,622.11	\$9,248.05	96.11%	\$26,696.71	\$18,870.16	\$7,826.55	41.48%
Week 2	Sunday	\$20,601.16	\$10,042.42	\$10,558.74	105.14%	\$22,746.88	\$20,601.16	\$2,145.72	10.42%
	Monday	\$17,260.99	\$11,521.88	\$5,739.11	49.81%	\$19,710.30	\$17,260.99	\$2,449.31	14.19%
	Tuesday	\$18,879.64	\$10,236.11	\$8,643.53	84.44%	\$25,830.88	\$18,879.64	\$6,951.24	36.82%
	Wednesday	\$19,650.25	\$13,706.01	\$5,944.24	43.37%	\$24,191.63	\$19,650.25	\$4,541.38	23.11%
	Thursday	\$21,339.78	\$10,374.02	\$10,965.76	105.70%	\$23,072.23	\$21,339.78	\$1,732.45	8.12%
	Friday	\$17,785.43	\$11,568.69	\$6,216.74	53.74%	\$22,086.22	\$17,785.43	\$4,300.79	24.18%
	Saturday	\$15,552.95	\$9,400.84	\$6,152.11	65.44%	\$27,105.86	\$15,552.95	\$11,552.91	74.28%

Conclusion

SSAS now has built-in time intelligence which developers can use to quickly and easily create time calculations. But manually creating "shell" time dimensions still offer the greatest flexibility when it comes to implementing global time calculations. Shell dimensions can also be used for other "global" type of calculations, such as mean, median, mode, etc.

Support Files:

[Sample Database and Project Files](#)

Other Resources:

[Defining Time Intelligence \(SSAS\)](#)

[Time Intelligence Wizard Fix](#) (If you don't have SP1)