CORPORATER express

STEP-BY-STEP TUTORIAL

Corporater Express Step by Step Tutorial

Introduction

While Corporater Express is designed for business users, it introduces some concepts that may be unfamiliar to new users. Prior to using Express, most users track metrics using a spreadsheet, such as Excel. This tutorial assumes you know your way around Excel. It illustrates how to start using Corporater Express, coming from Excel.

Basic Concepts

Spreadsheets-Cell-based data

Excel uses a cell-based approach to managing and storing data. This means that only one data record can occupy a cell at a time. Data is generally handled using the two dimensions of column and row. This enables people to very quickly organize data and easily make basic calculations, generally in table form.

> When creating *scorecards*, the limitations of spreadsheets quickly become apparent. It can be challenging to maintain links to data contained in other worksheets and workbooks. Extra work is required to create *KPI* pages that display additional data, charts, and graphs. Navigating to different organizations or time periods can be difficult. Most spreadsheets are not web-enabled. Data collection tends to be manual, and poses challenges if there are multiple data providers. There are few security and access control features available.

Corporater Express - Data records

Corporater uses a data record approach to store *scorecard* data in a database. Data records offer the benefit of easily accommodating multiple data records for a given time period. This makes it very easy to change the time period for reporting data and metrics. For instance, you can easily change between a monthly and quarterly view of the data. To use data records in Corporater Express, you first need to build a basic data structure. This allows you to "*drill-down*" from high level data into source level data quickly and easily.

Corporater Express is designed specifically as a performance management *scorecard*. You can quickly and easily navigate to other reporting periods and organizational units. *Scorecards* can link to data contained anywhere in the organization. Since it is web-based, it is easy to manage multiple users and data providers with a high level of control and security. Tools and workflow have been developed to support both automatic and manual data collection. Commonly used elements, such as charts and graphs, are pre-built and are enabled by simply selecting them from a menu.

Nodes - Data record containers

Nodes are the basic elements for creating a data structure. They are data "containers" or categories, such as revenue, number of hours worked, sick time, number of open positions, etc. *Nodes* can be created hierarchically. Nested under revenue, you can list revenue sources, types, or categories. The result for revenue will automatically be the sum of the *nodes* nested beneath. This enables you to keep your *nodes* organized, as well as to provide paths for "*drill-downs*." Additionally, *nodes* offer sub-types, such as *actual*, *budget*, *target*, and *forecast*. You do not need to build separate *nodes* to show *budget* figures. Your *node* structure will become important as you start to aggregate data. You don't need to spend much focus on your structures as you are learning the software, since you can always reorganize *nodes* later.

To access data from *nodes*, we need to know the *node(s)*, *organization(s)*, and the *time period* of the data, rather than spreadsheet column and row. A simple calculation takes the data from one *node* and one *organization* (selected by the *scorecard*). The *time period selector* determines the time period. To add data from multiple *nodes* and *organizations*, we use the aggregation function. Aggregation is how company-wide data can be calculated and displayed, for example.

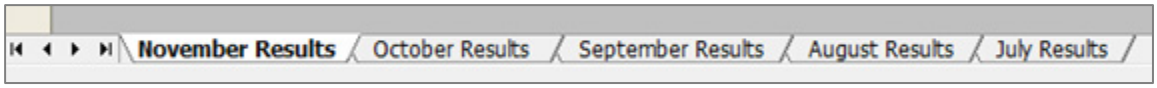
Understanding the differences

Excel approach

	A	B	C	D	E	F
1	**Corporate Balanced Scorecard - November 2008**					
2	FINANCIAL				CURRENT PERIOD	
3	**Increase revenue**		**Responsible**	**Target**	**Actual**	**Indicator**
4	Revenue		John Grossbard	83.5	82.6	G
5	Expenses		Ben Cantwell	79.1	80.1	G
6	**Profitability**		**Responsible**	**Target**	**Actual**	**Indicator**
7	EBITDA		John Grossbard	4.4	2.4	G
8	Profit		Ben Cantwell	5.3	3.0	G

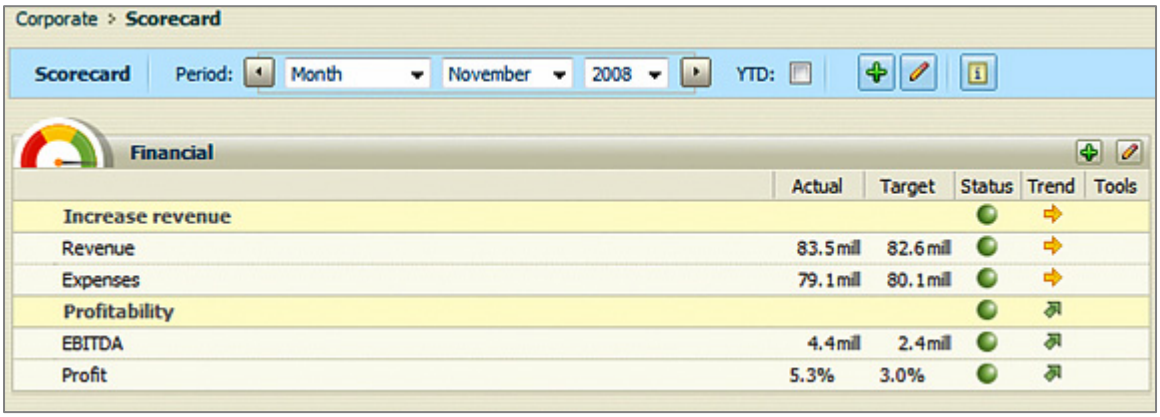
The cell E4 contains either the *actual* value for revenue, or a formula for calculating revenue that links to other cells in the spreadsheet or workbook. These links can become difficult to manage as the complexity of the *scorecard* increases over time. Additionally, you cannot easily *drill-down* into the data to see where the revenue is coming from (by either revenue type, or by organizational unit).

Time navigation



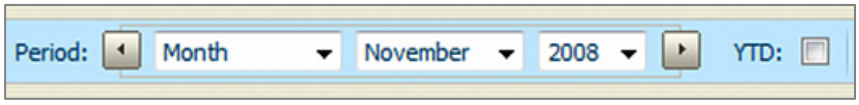
Each new month requires creating a new *scorecard* worksheet. It is difficult to change reporting periods to quarter, or to switch to a year-to-date view, unless that view has been specifically built in the spreadsheet.

Example of a *scorecard* in Corporater Express



By clicking on revenue, the user is taken to a *KPI* page that can display additional underlying data. A *drill-down* table can show how different revenue types and different organization units contributed to the *actual* revenue figures for the month. It is very easy to add monthly charts and graphs that illustrate the monthly performance, and to add other supporting data.

Time navigation



The *time period selector* determines the reporting period, and can switch to quarterly results or year-to-date results. No additional calculations are required when changing periods.

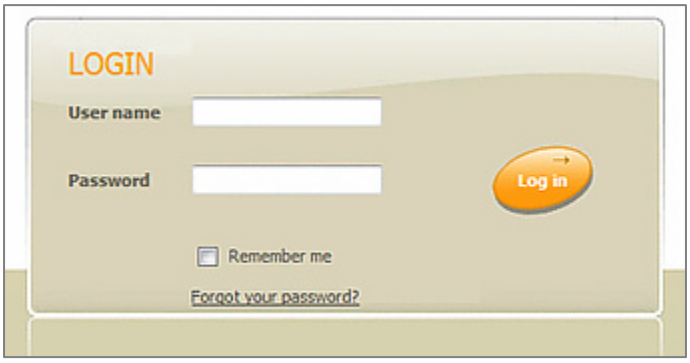
BASIC TRAINING

Concepts covered

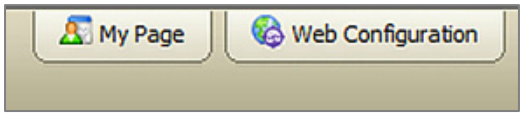
1

Preparing your workspace

1. Log in to your Express workspace

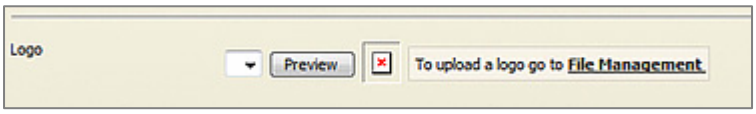


2. Select the *Web Configuration* tab in the upper right corner

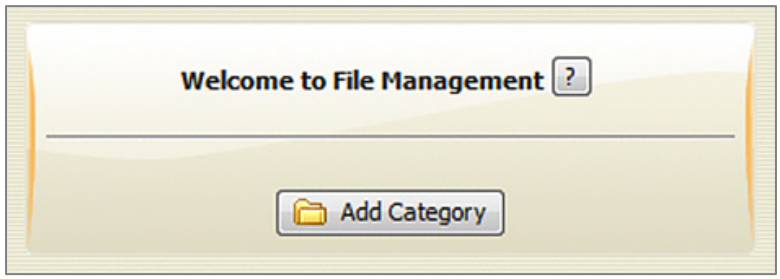


3. Begin by adding an *organization*

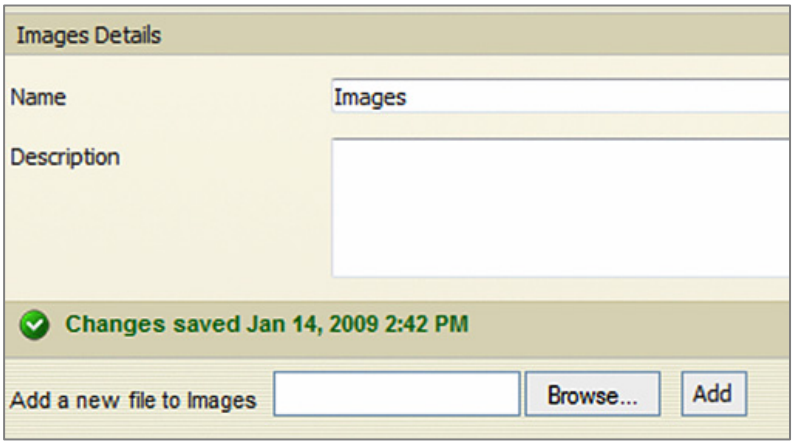
4. Upload a jpeg or gif image (no larger than 150 X 50 pixels) - go to *File Management*



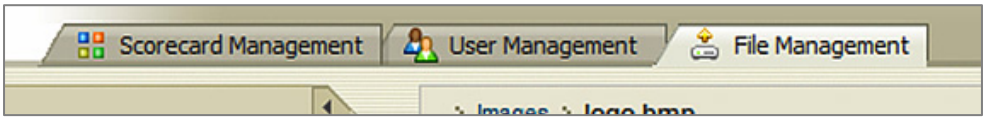
5. At *file management* create a Category (folder- ie. images)



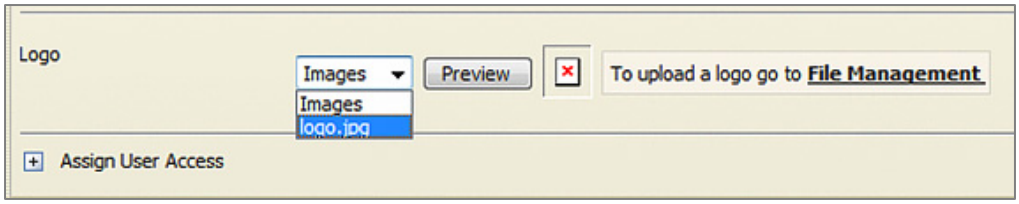
6. Upload your logo by selecting Browse, then Add



7. Return from *File Management* to *Scorecard Management*



8. Select and save your logo (all “children” of the organization can share the same logo, if desired)



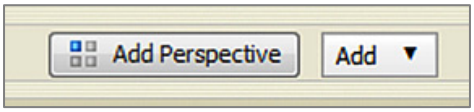
2

Creating a *scorecard*

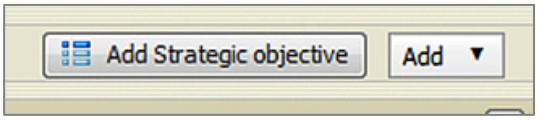
1. Select *Add Scorecard* in upper right corner (or add additional organizations)

Name your *scorecard* (to change it from the default name). You can also add vision, missions, and main goal.

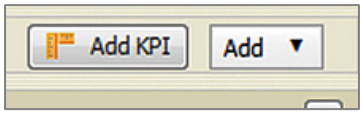
2. When your *scorecard* is selected, you can add a *perspective* by selecting it in the upper right corner. In following our Excel example, we will name it Financial.



3. Continue by adding a *Strategic objective*, for example Increase Revenue



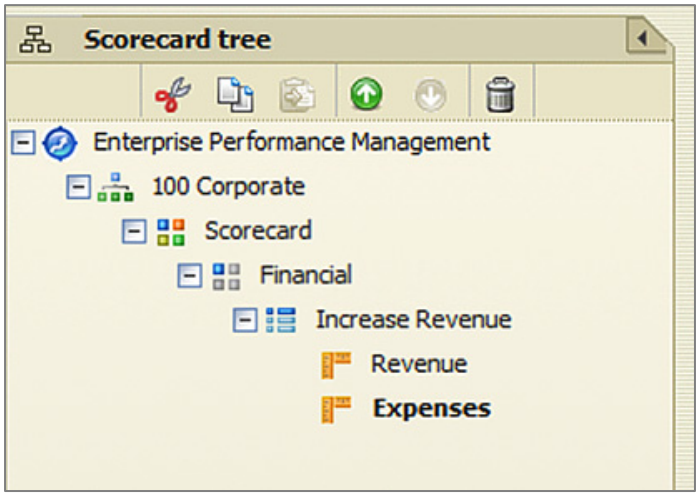
4. Add a *KPI*, such as Revenue



Don't forget to press save after every step. Depending on your screen resolution, you may have to scroll down to access the save button.

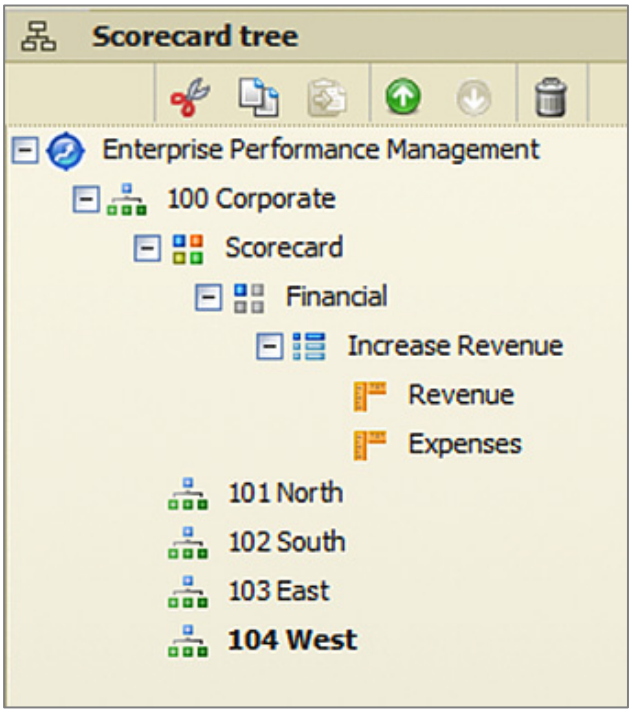
Summary

This is how we build the *scorecard* structures. To add another *KPI*, select the *Strategic Object* above Revenue (for example Expenses). Likewise, to add another *Perspective*, select the *Scorecard* itself.



So far we are building *scorecards* and *organizations*. To add another organization unit under Corporate, select Corporate and *Add Organization* in the upper right corner. To illustrate functionality, we will add North, South, East, and West under Corporate.

Our *scorecard* now looks like this:



Keep in mind that we have just created organizations, and they do not yet have their own *scorecards*.

3

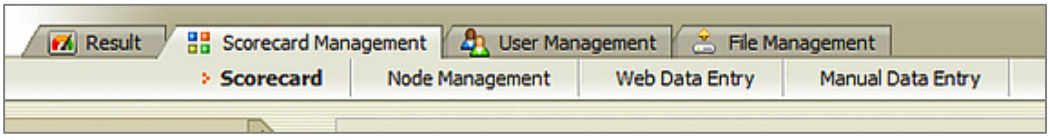
Creating a *KPI*

Everything you add under the *KPI* can be shown on its own *KPI* page. The page is accessed from the main *scorecard* page by clicking on the name of the *KPI*. The *KPI* page can display underlying data, monthly tables, charts, and graphs, comments, task lists, check boxes, etc.

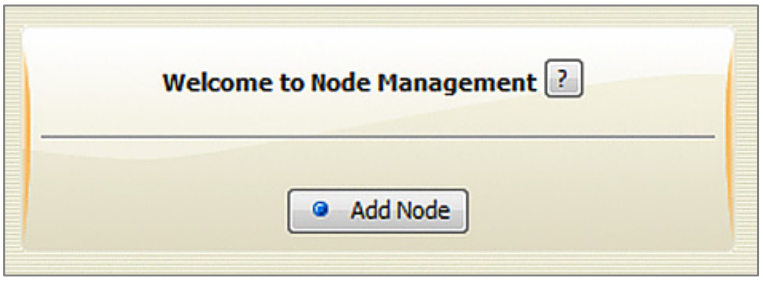
*KPI*s generally require data, therefore we need to create a basic data structure.

Creating a *node* structure

1. Select *Node Management* on the *Scorecard Management* sub-tab



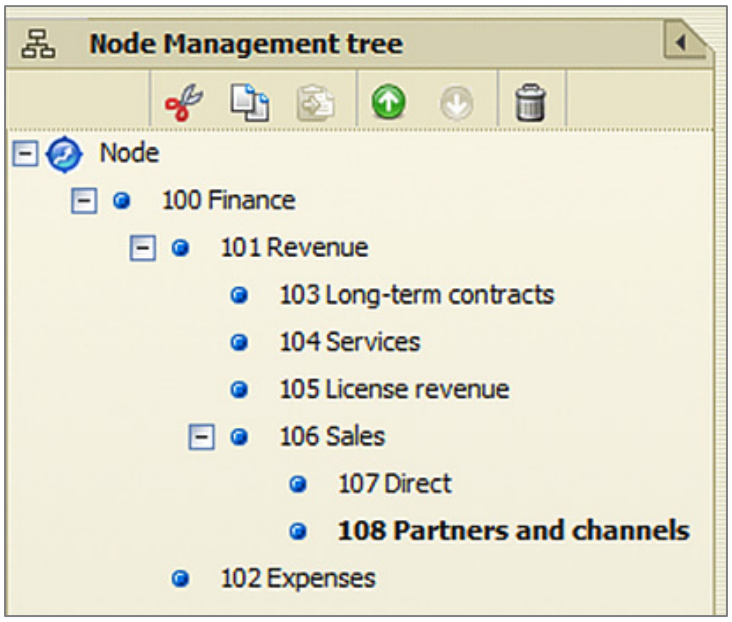
2. Begin by adding a *node* - we will name it finance, and only use it to store other finance related *nodes*.



3. Note the formatting options. The formatting options on this page will be used to format the data in *drill-down* tables. You can select different formatting options for how data is displayed in the *KPI* functions (the results of calculations).

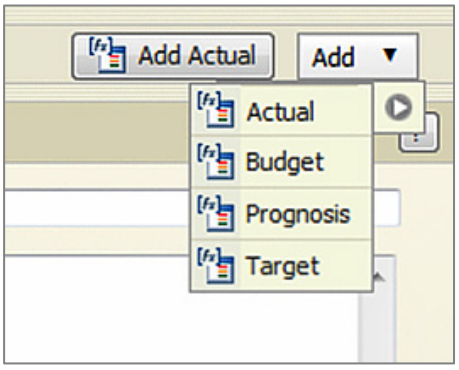
Nested *nodes* will default to the formatting of the parent *nodes*. If you plan to build a deep *node* structure that shares common formatting, start by selecting the formatting at the parent *node* level. All child *nodes* will share the same formatting.

While Finance is highlighted in the *Node Management* tree, select *Add Node* in the upper right corner. In our example, we will name it Revenue and use it to hold nested *nodes*. A very simple nested structure has been created, as follows:

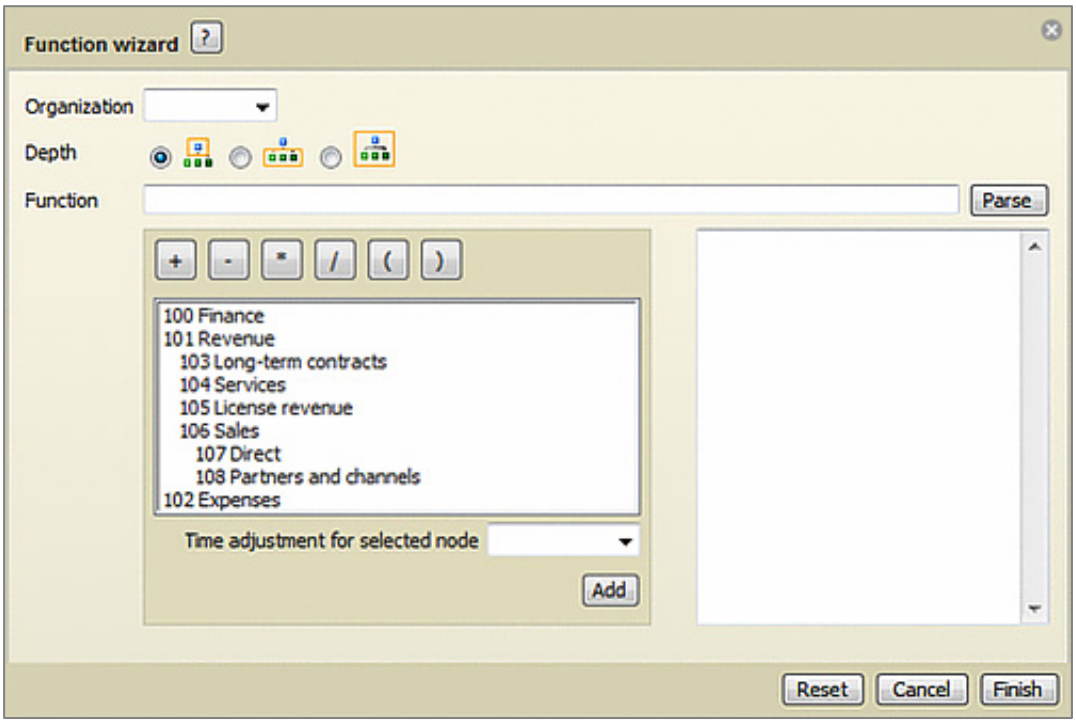


4. Navigate back to the *Scorecard* sub-tab beneath *Scorecard Management* to enable editing of your *Scorecard* elements. Select the *KPI* Revenue. Add a *simple status* in the upper right corner. The *simple status* will hold the threshold values that determine the color of the *status lights* on the *scorecard*. We will come back to that later.

   The *Status* is necessary to hold the *Function*. The *Function* is where the value for the *KPI* is generated. It is similar to the Function bar in a spreadsheet. It uses its own syntax and offers a variety of mathematical functions. Selecting *Actual* will base the calculations on the *Actual node type*.



5. Select *Edit function* to activate the *function wizard*.

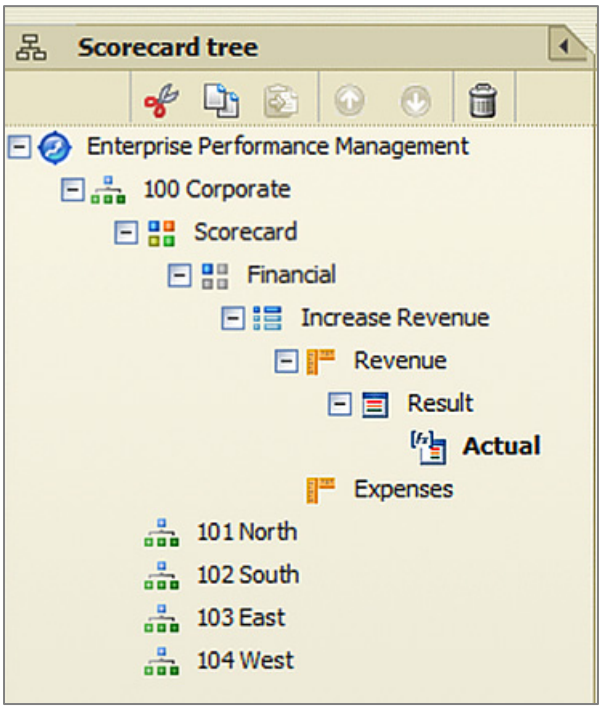


Functions

Standard

There are two main types of functions: standard, and aggregation. If we simply select the Services *node*, our function will appear as follows:

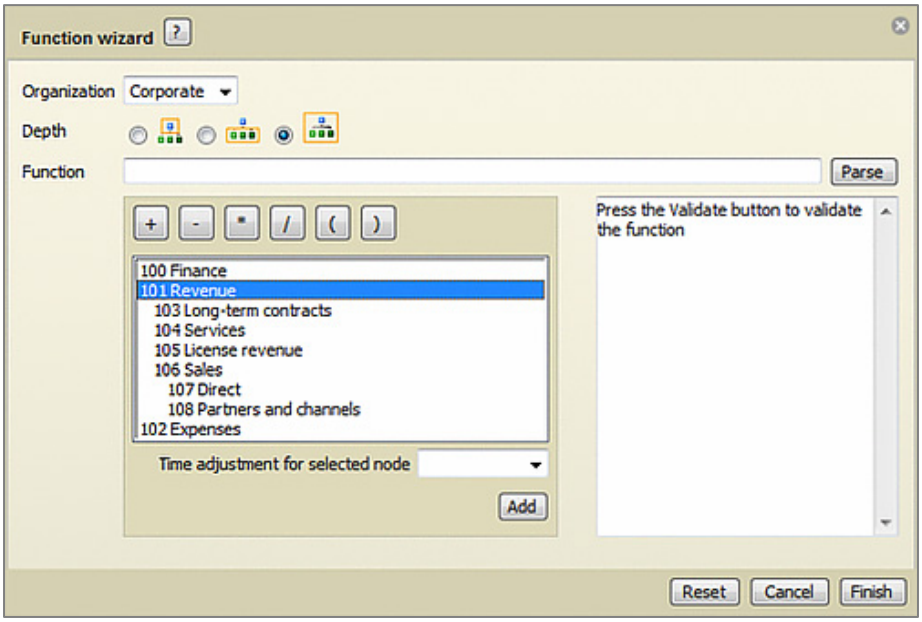
In this particular instance, it will provide the value of *node* 104 (service revenue) that is stored under organization currently selected (in this case Corporate). It will provide the sum of all values for the time period selected (month, quarter, etc.).



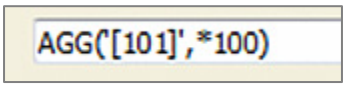
Aggregation

To show the sum of multiple organizations, such as all company revenue, we use aggregation.

We go back into our function editor to pull up the wizard. In this case, I select the organization Corporate, and the third *depth* option (of all units relative to Corporate). From the Revenue *node*, I select Add.

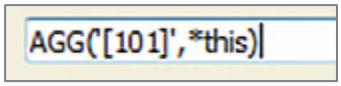


My function now looks like this:

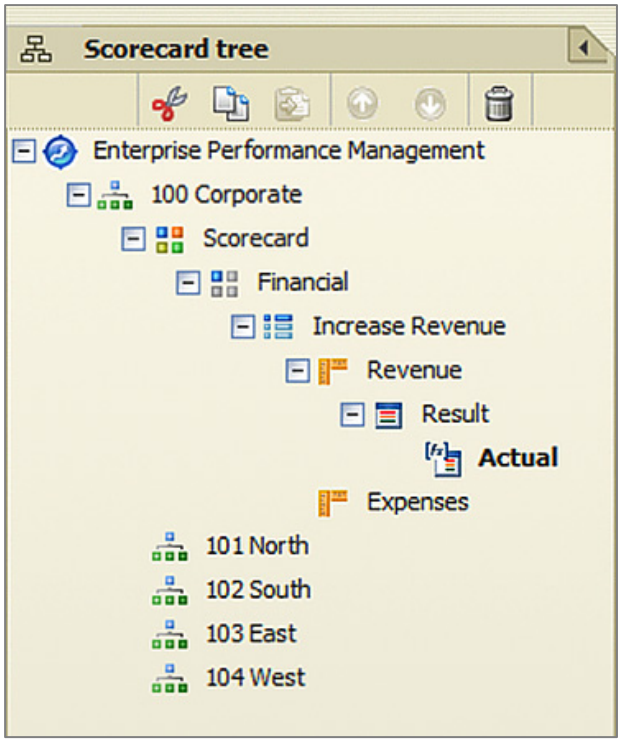


In this case, I will return the value that sums the Revenue *node*, including all the child *nodes*. It will include the Corporate, North, South, East, and West organizational units. It will include all the data from the time period selected in the *scorecard*. If I were to use this formula in any other *scorecards*, it would return the same value, since it is directly linked to the Corporate and all child *scorecards*.

We could also build a function that returns the same value using the "*This*" feature. By selecting "*This*" rather than Corporate in the organization selector in the wizard, we end up with the following function.



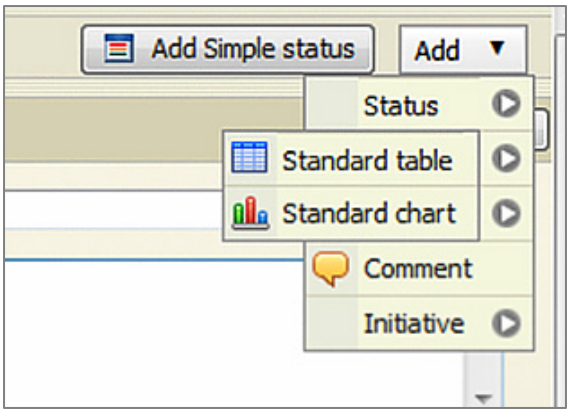
"*This*" enables us to select organizations relative to where the *scorecard* is used within the organizational tree.



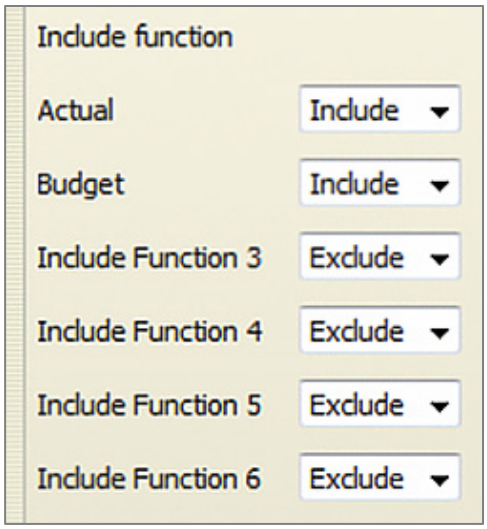
If we were to copy and paste the *scorecard* into the child organizations (North, South, East, and West), our function would only return the revenue values for each respective organizational unit. This offers an easy way to re-use functions throughout your organization that only includes information relative to the *scorecard* where it is used.

If we move back up one level (to the *Simple Status* object), we can add a *budget* function. This is done in the same manner as how we created the *actual* function.

If I select the Revenue *KPI*, I can now add charts and tables that reflect the values that are calculated. This is done by using the *Add* button, and then selecting *Table and Chart* under the Standard menu.



Both charts and tables offer the option to select included functions



In our case, we only have two, and we will select both.

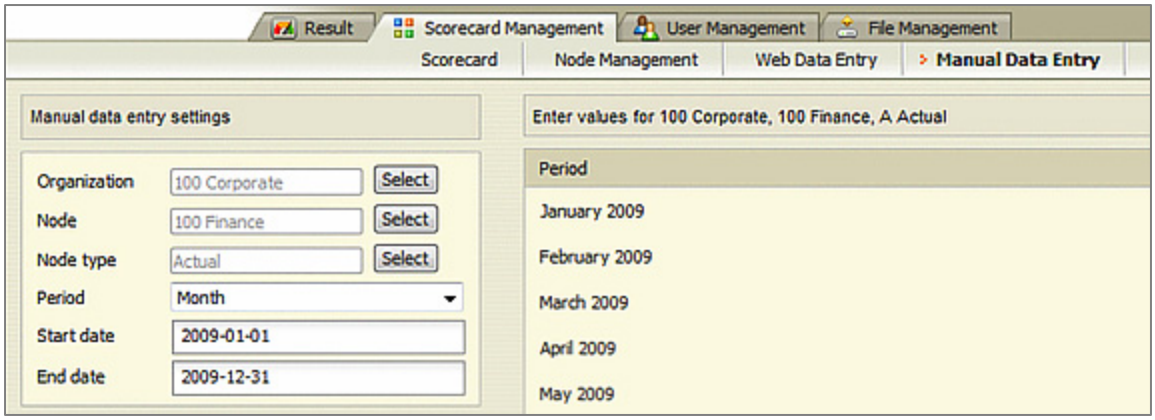
Summary

We have now created a *KPI* with charts and tables. We have a basic data and organization structure. All we need now is some data so we can view our results.

4

Adding data

The easiest way to add data is through *Manual Data Entry*



Select the *organization*, *node*, and *node type* (*Actual, Budget*, etc) to enter your data. In our case, we need to add data under Revenue. Enter your data for several of the *nodes* nested under revenue. You can also select different organization units.

NOTE

A note about data records

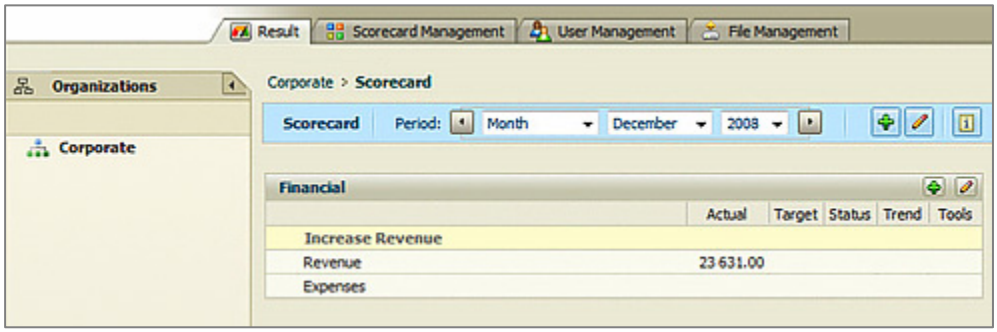
If I were to make another *manual data entry* of 1000 in August, the sum would be 50,087. The 1000 does not overwrite the previous entry, but rather is added to it.

5

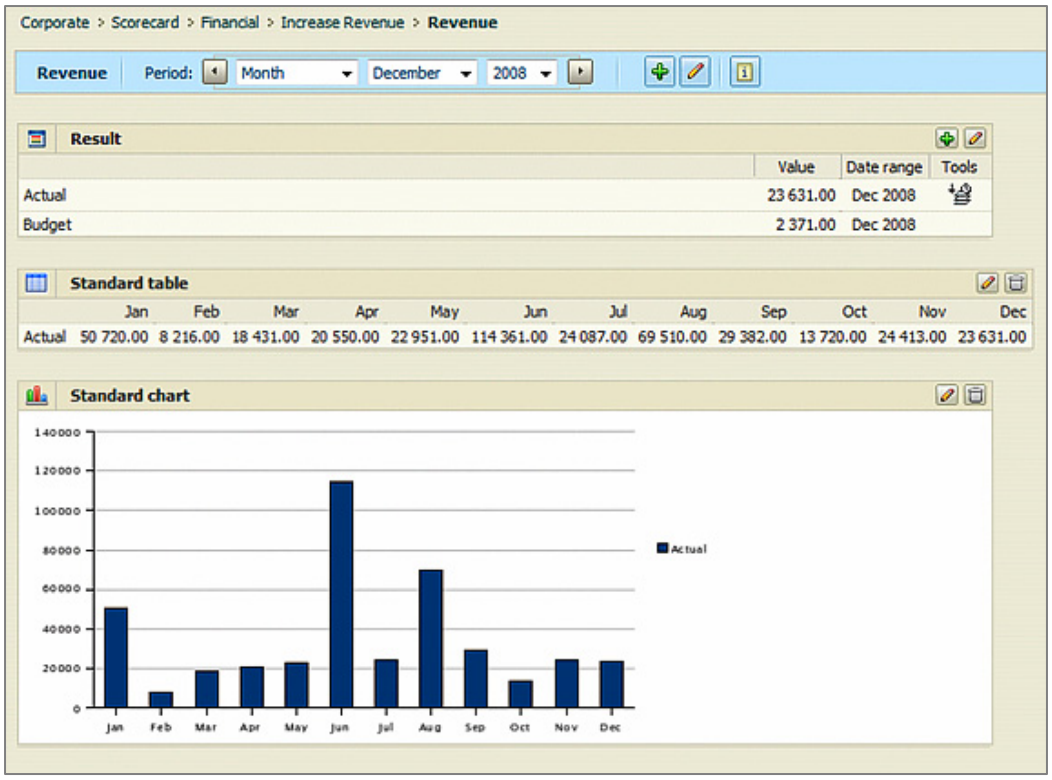
Working with *Scorecards*

Scorecard view

So far our *scorecard* looks like this:



The organizations North, South, East, and West are not visible because they are not linked to any *scorecard*. But we can still add data to those organization units. If we click on revenue, we arrive at the *KPI* page.



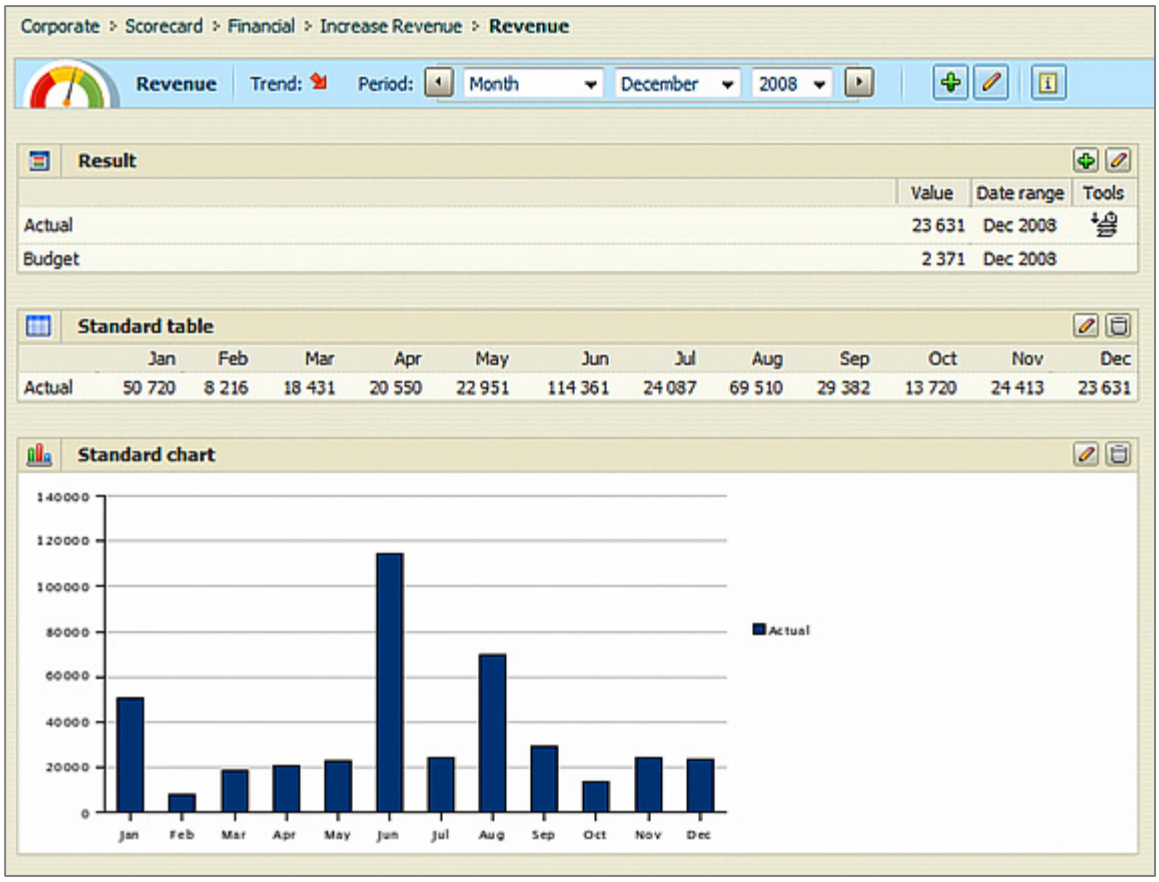
In the example above, it reflects data entered manually. It would probably make sense to eliminate the decimals in the table (edit formatting). Also, the tables can be renamed to something more descriptive. Finally we have not yet selected *status* thresholds to determine the color of *the status lights* or *meters*.

Setting *Status* Thresholds

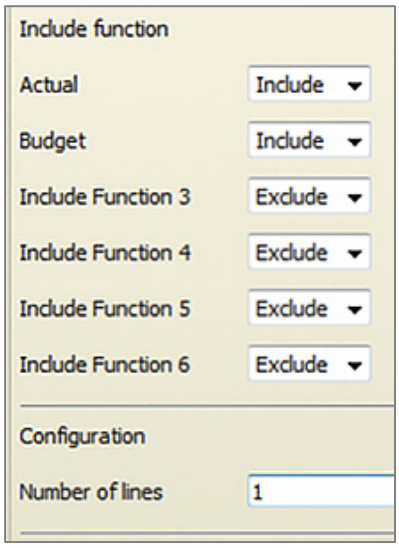
Thresholds are set by selecting the *Simple Status* object in your *scorecard* tree view. By default it is named *Result*.

Please note that the commas were not entered, but were rather formatted after the threshold was saved. In the above example, anything above 30000 will show maximum green on the *speedometer* (needle pointing all the way to the right). 25000 will be yellow, and the needle will be exactly between yellow and green. Similarly 20000 will be between yellow and red, and anything lower than 15000 will be maximum red.

Our *KPI* view now looks as follows:

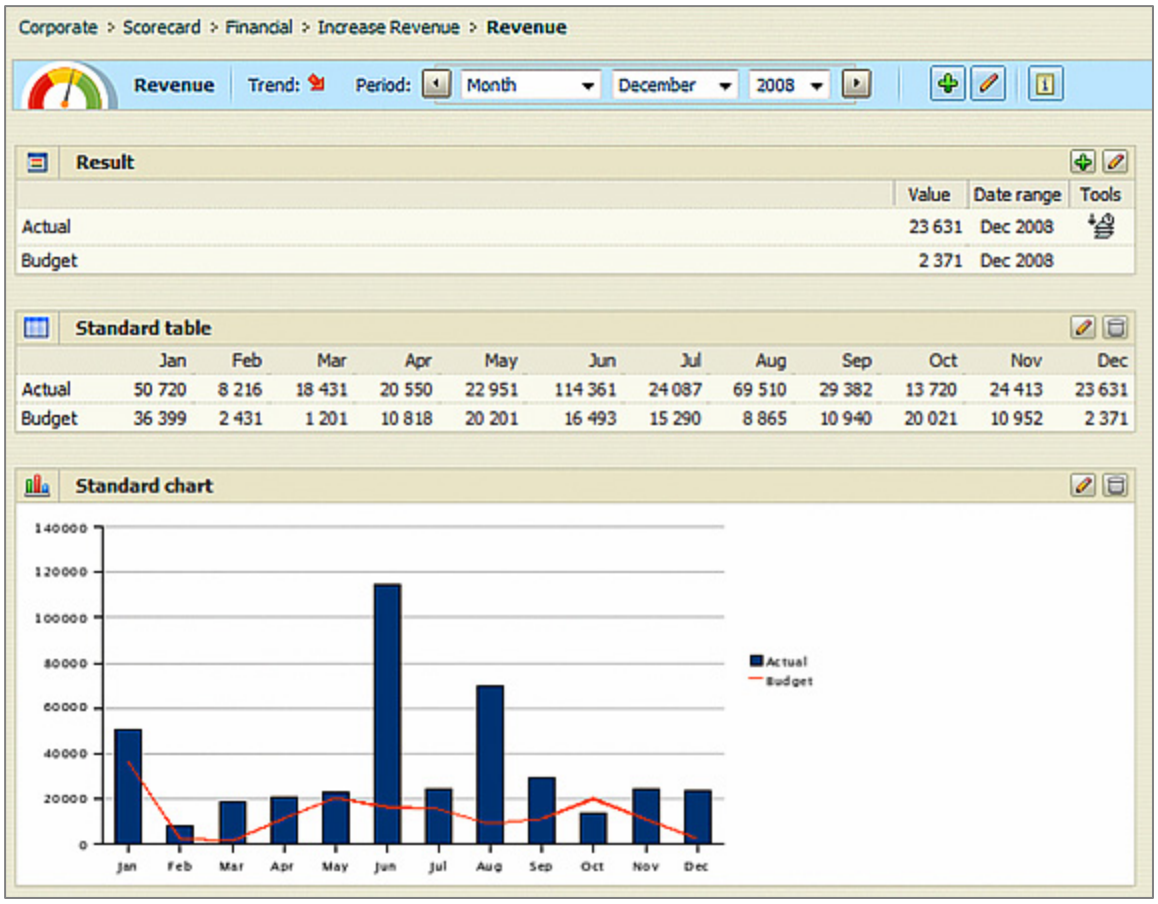


To add the *budget* values to the table and chart, we select those in their properties box.



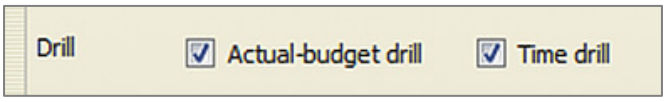
By selection one line, the *budget* will be shown as a line, and *actual* as a bar. If I select 2 lines, both would be shown as lines in the chart.

Our *KPI* view now looks as follows:



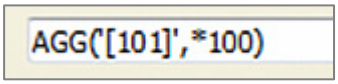
Drill Down

On your *actual* and *budget* details page, you can select two *drill-down* options:



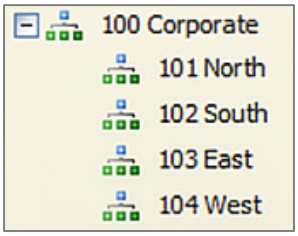
The *drill-down* page is accessible by clicking its icon, under tools next to *actual, budget*, etc.

Actual-budget compares the two values and calculates the variance. *Time drill* offers a monthly view of the expanded data. Your *drill-down* structure will follow the same structures as your *nodes* and *organizations*. Example: the following function is used in a *drill-down*.



This aggregates the *node* 101 (and all *sub-nodes*), through the organization 100 and all of its children. In more specific terms, *node* 101 is revenue, and organization 100 is the top level organization. The asterisk was generated when we selected all child organizations. So in this case, all organizations in the company will be included.

Our *organizations* that will be included are as follows:



Our *nodes* that will be included are as follows:

In the example shown, there is only data entered under the organization North, and under the *nodes* Long-term contracts, Services, License revenue, Direct, and Partners and channels. The Sales *node* is a placeholder, as is Revenue.

As we select a time-based *drill-down*, this is what we see:

Time drill					
Name	**January**	**February**	**March**	**April**	**May**
Corporate	50 720.00	8 216.00	18 431.00	20 550.00	22 951.00
North	50 720.00	8 216.00	18 431.00	20 550.00	22 951.00
South	N/A	N/A	N/A	N/A	N/A
East	N/A	N/A	N/A	N/A	N/A
West	N/A	N/A	N/A	N/A	N/A
Name	**January**	**February**	**March**	**April**	**May**
Revenue	50 720.00	8 216.00	18 431.00	20 550.00	22 951.00
Long-term contracts	23 432.00	343.00	N/A	456.00	654.00
Services	9 876.00	76.00	798.00	68.00	4 768.00
License revenue	868.00	354.00	75.00	3 463.00	875.00
Sales	16 544.00	7 443.00	17 558.00	16 563.00	16 654.00
Direct	6 866.00	876.00	8 769.00	8 687.00	7 865.00
Partners and channels	9 678.00	6 567.00	8 789.00	7 876.00	8 789.00

You can see that even though there is no data under the Sales *node*, it holds the value from adding the child *nodes* Direct, and Partners and channels. Likewise, Revenue sums all the data contained beneath it.

At the organizational level, Corporate holds no data, but rather adds the data of all the child units. In this example, North is the only organization holding data. Corporate could contain its own data, if needed. Likewise, the placeholder *nodes* Direct and Revenue can also contain their own data. Your own business case will dictate how you handle *nodes* like these.

Actual-budget drill

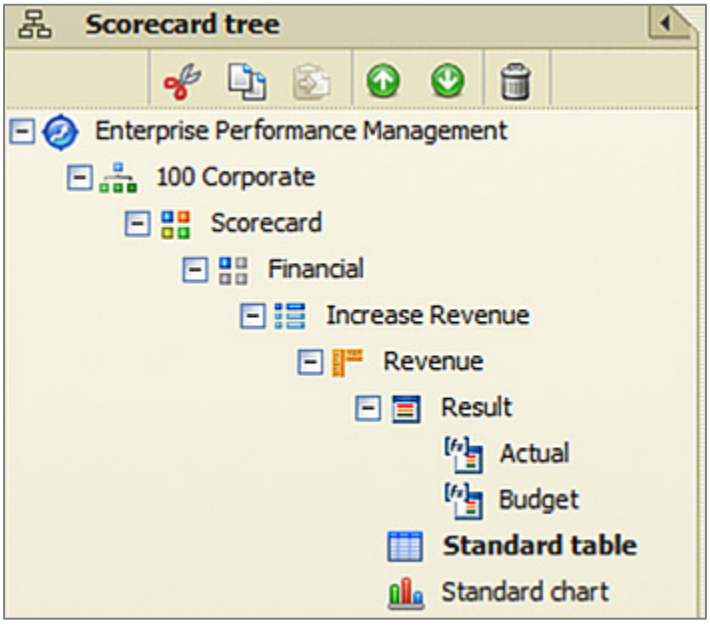
Actual budget drill

Name	Actual	Budget	Deviation	Actual YTD	Budget YTD	Deviation
Corporate	23 631	2 371	21 260	419 972	155 982	263 990
North	23 631	2 371	21 260	419 972	155 982	263 990
South	N/A	N/A	N/A	N/A	N/A	N/A
East	N/A	N/A	N/A	N/A	N/A	N/A
West	N/A	N/A	N/A	N/A	N/A	N/A

Nested *nodes* will default to the formatting of the parent *nodes*. If you plan to build a deep *node* structure that shares common formatting, start by selecting the formatting at the parent *node* level. All child *nodes* will share the same formatting.

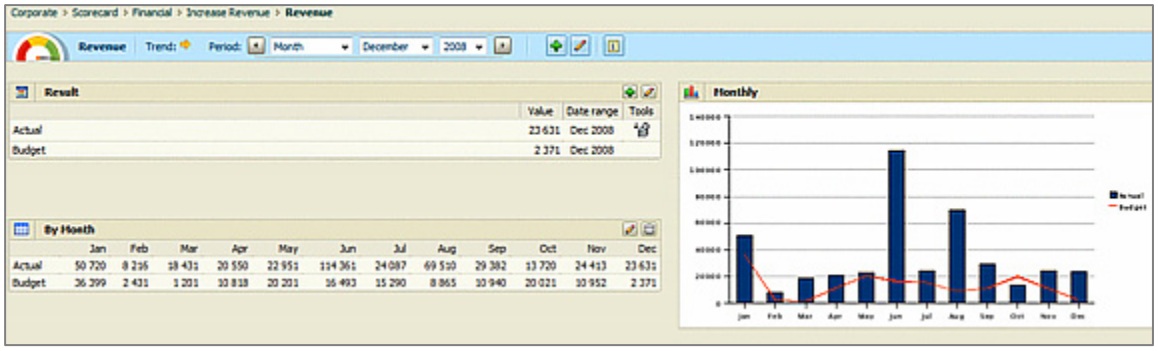
Page Layout

Elements on any *scorecard* page can easily be formatted for the web. By selecting an object and the up/down arrows, their order can be rearranged.



By using row, column span, and width, you can rearrange objects so they are next to each other.

By changing all elements to have a column span of 2 (out of 4) and a width of 100 %, we create two columns. The *Standard chart* is moved up and renamed. It is given a row span of 2, meaning that it allows room for two objects next to it, resulting in this new layout.



Definitions

Data Record

Data contained by a *node*. Data can be entered manually through *Web Data Entry*, or through the *Manual Data Entry* under *Web Configuration*. Automatic data collection is also available, as are *web forms*.

Please note that to change the value of a data record, you need to delete the previous value first, otherwise you will be adding a new data record that will be added to it.

The exception to this rule is that you can overwrite your own *Web Data Entry (WDE)* records from within the *WDE* interface.

This is a feature designed to provide control over the unauthorized altering *scorecard* data.

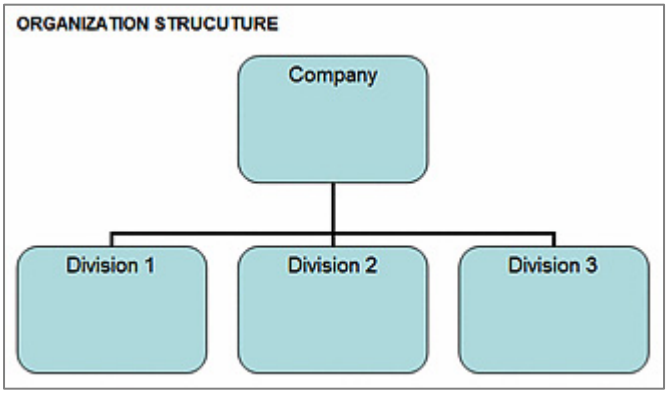
Organization

A business unit that has a *scorecard* and/or data linked to it. Organizations are typically created in a hierarchical structure. There may be a top level *scorecard*, with several sub-units beneath. There are no limits to how deeply organizations can be nested.

You may find it practical to create organizations for the purposes of keeping your data organized, even if those organizational units have no *scorecards* or metrics of its own. You may also create organizations for the purpose of managing your data. This will enable you to *drill-down* into data attached to that organization. For instance, if you maintain service level agreements with a few large customers, you can create organizations representing customers. This will enable you to who customer satisfaction or complaints for those categories.

Basic Concepts

Parent-Child Relationship



In an organizational structure, it can be said that Division 1 is a “child” of the parent organization “Company,” or that Divisions 1, 2, and 3 are all “Children.” Similar relationships can be built with *nodes*. These relationships become significant when “aggregating” data.

Data Aggregation

Large amounts of data may be summarized by using aggregation, for example, monthly results for sales figures. Using the example above, you can display data by each division for comparison purposes, and sum them to show total sales for the company. Also, you could display data by the type of sales, such as product or service type. Aggregation by data type follows the *node* structure. Corporater Express contains functionality that automates the aggregation and features such as *drill-downs*. To fully benefit from data aggregation, it is important to understand how *node* structures function.

Nodes

A Business Case

Consider the case of a business that is looking to track payroll and overtime. The business in comprised of three divisions. Each uses a combination of permanent employees and temporary staff. Their payroll is comprised of regular work hours, overtime (at 1.5 times the normal pay rate), and holiday pay (at double the regular hourly wage). The company sometimes budgets hours for overtime to address special projects, but this is not always the case. Due to the type of business (health care, for example), they have service level agreements requiring them to maintain a certain staffing ratio. Overtime is variable based on unfilled positions and unexpected work absences (usually due to illness).

Spreadsheet model

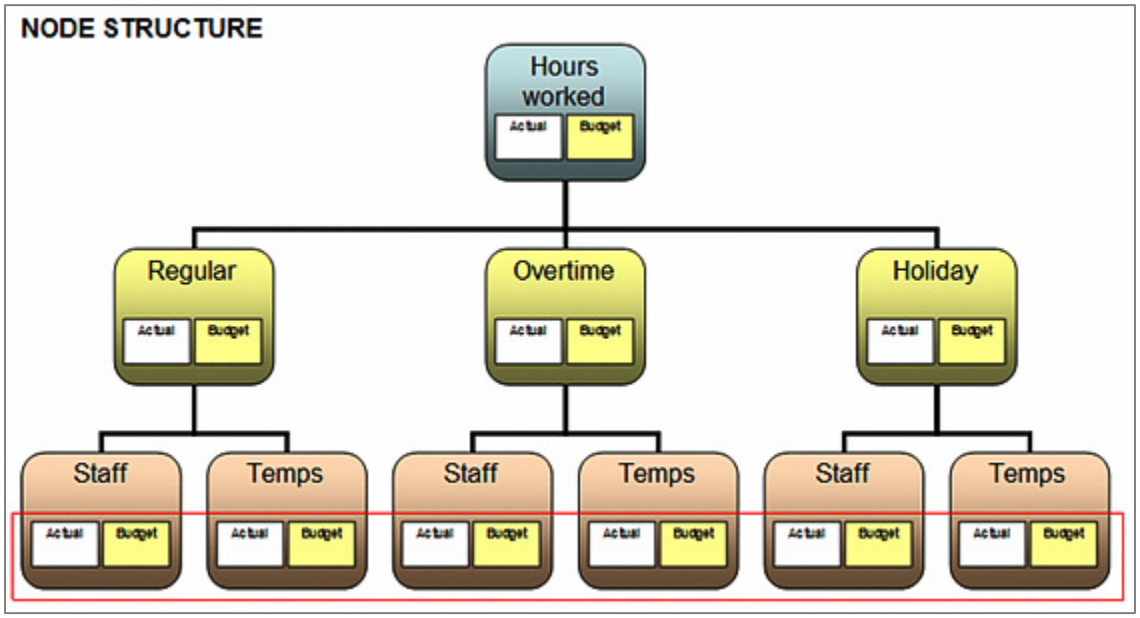
Month of January

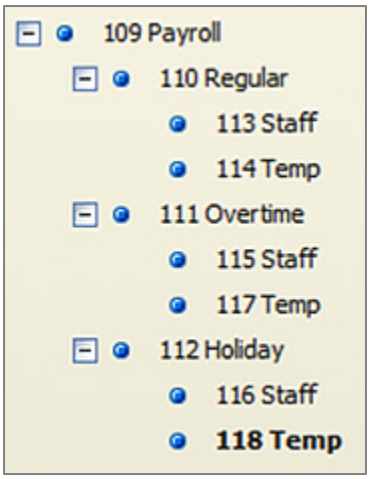
	A	B	C	D	E	F	G	H	I	J	K
1											
2											
3											
4		PAYROLL		Division 1		Division 2		Division 3		Corporate (total)	
5				Actual	Budget	Actual	Budget	Actual	Budget	Actual	Budget
6		Regular									
7			Staff	$200,182	$240,000	$150,780	$160,000	$280,876	$320,000	$631,838	$720,000
8			Temps	$40,000	$0	$12,987	$0	$26,784	$0	$79,771	$0
9				$240,182	$240,000	$163,767	$160,000	$307,660	$320,000	$711,609	$720,000
10		Overtime									
11			Staff	$35,984	$20,000	$12,012	$20,000	$17,283	$0	$65,279	$40,000
12			Temps	$256	$0	$0	$0	$218	$0	$474	$0
13				$36,240	$20,000	$12,012	$20,000	$17,501	$0	$65,753	$40,000
14		Holiday									
15			Staff	$0	$0	$0	$0	$0	$0	$0	$0
16			Temps	$0	$0	$0	$0	$0	$0	$0	$0
17				$0	$0	$0	$0	$0	$0	$0	$0
18		TOTAL		$276,422	$260,000	$175,779	$180,000	$325,161	$320,000	$777,362	$760,000

In this example a cell needs to be created to calculate and display each data element. For example, to calculate how much overtime Division 2 used for their permanent employees, you will find it in cell F11. To add up all the overtime of Divisions 1, 2, and 3 for regular employees, you will need to add it up using a formula as follows: =D11+F11+H11. Spreadsheets offer a quick and easy way to display simple data.

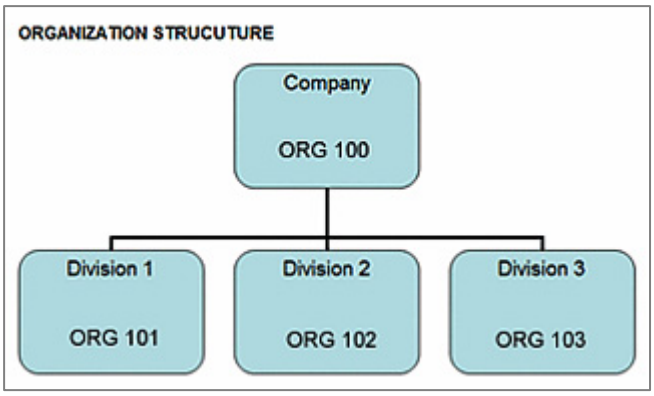
Node Model

The *node* model of displaying data offers richer functionality and options for navigating through large amounts of data. Data is organized by data type, the organization it is related to, and the time period. In this case, we build a simple *node* structure where Regular, Overtime, and Holiday are all nested under Hours worked. Staff and Temps are added to each *node* to reflect the different employee categories. *Actual* and *Budget* are default *node type*s, and require no action on the user to create.





In this business case, only the *nodes* above that are marked in red (numbered 113-118 in the *node* tree) contain the *actual* data. In this example, the *node* "Overtime" contains no *actual* data, but by aggregating it sums the data of the child *nodes*: the Staff and Temp data. You can also place data in Regular, or Hours worked, and when aggregated, that data will be added to the data of the child *nodes*.



Summary

Data is organized by data type, the organization it is related to, and the time period.

Functions

A simple function looks something like this: [101]. The *node type* is also selected, and by default is *Actual*. This function takes the *actual* value for the *node* with the ID #101, for example, "customer complaints." This function still needs to know what organization and time period to report on. It will derive that information by the *scorecard* where the function is used, and from the time period that is selected. For example, if the function is used in the *scorecard* on Division 2, and there is one complaint registered each month, it will display the value of 1 for a monthly view, and 3 for a quarterly view.

Aggregation

To add up the values for a given time period for multiple organizations, it would use aggregation.

FORMAT

Using *period selector* - AGG('[*NODE*]', ORGANIZATION) or Custom time period - AGG('[*NODE*, BEGINNING OF PERIOD, END OF PERIOD]', ORGANIZATION)

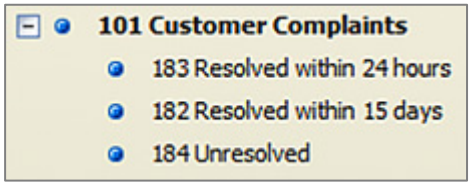
The function AGG('[101]',*this) or AGG('[101]',*100) would both yield the same result if used on the *scorecard* for Company (ID 100). In this case [101] refers to the *node*, and the *this or *100 refer to the organizations to aggregate the data. "This" is a relative way to express "this organization." In this case, the addition of the * means that it will sum the parent and all children of any organization where the function is used. Therefore, if the same function were pasted into the Division 1 *scorecard*, it would sum the value of Division 1 complaints, and the values of any children of Division 1 (none in this case).

*100 will sum the parent and children specifically of organization 100 (for example Company, Division 1, 2, and 3). It will deliver the same result regardless of where it is used within the organization, since the values are locked to organization 100 and all of its children.

In these cases, the *time period selector* determines the reporting period. Custom time periods can also be created. For example AGG('[101,BOY, EOP]',*100) will aggregate the year-to-date number of complaints for all organizations within the company. BOY refers to "beginning of year" and EOP to "end-of-period." Express offers a rich set of custom time functions that can be used to create very advanced calculations. These functions are beyond the scope of this tutorial.

It should also be noted that aggregating sums the values of the aggregated *node*, plus the values of all *child-nodes*. In the case of complaints, a company might break their complaints down by how they are resolved, or they might store all complaints in one *node*.

AGG('[101]',*100) will calculate the number of customer complaints for all organizations.

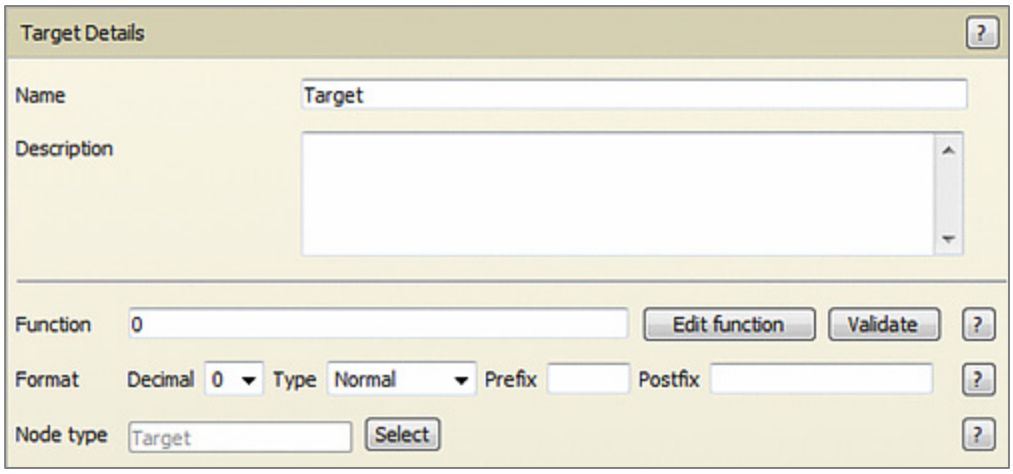


AGG('[101]',*100) will add up all the complaints for *nodes* 101, and 182-184. Additionally, a user will be able to *drill down* through the *node* structure to view how many *nodes* were unresolved for a particular business unit.

*Node type*s

By selecting *Target, Budget*, or *Forecast* you can make comparisons or calculations against other *node type*s. While this is not a practical example, it illustrates how you can calculate complaints be comparing them against your target.

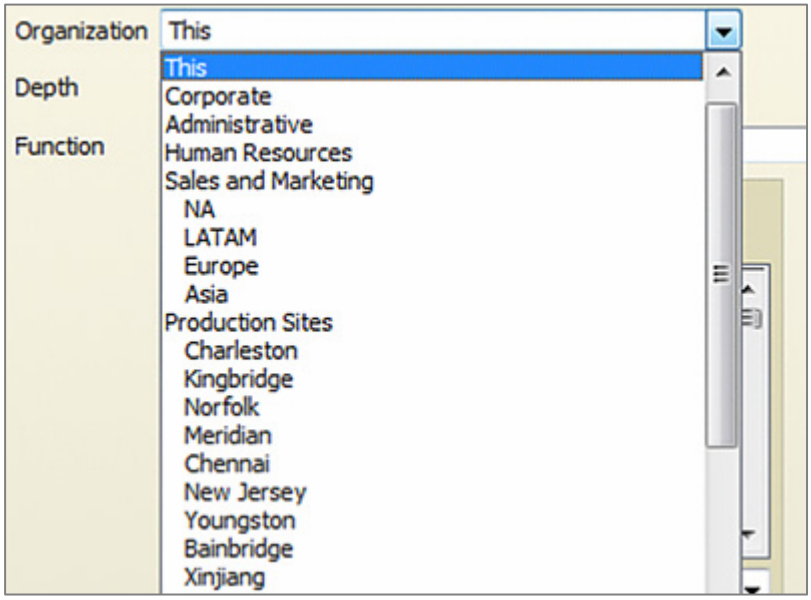
AGG('A[101]-T[101]',*100) This subtracts the target number of complaints from the *actual* number of complaints to calculate how far above target performance was. As this is likely not a practical example for this application, it is worth noting that you can simply enter the target value into the function box. For example:



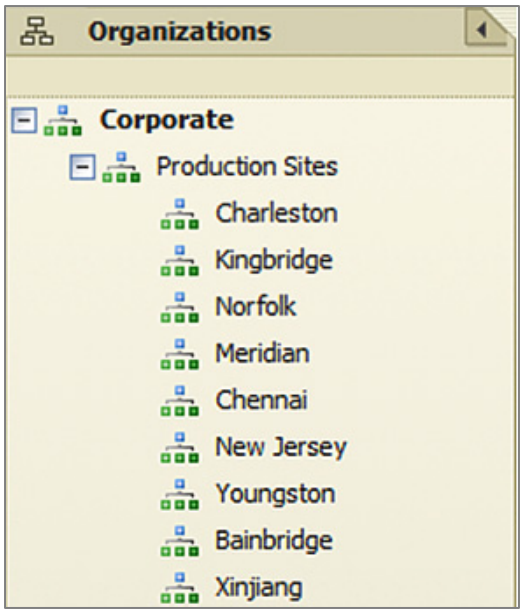
This example will show the *Target* value of 0 for each month. Since there is no *actual* data stored in a *Target node*, you cannot use T[101] in a calculation. However, since you already know its value, presumably you do not require the data in *node* form.

Selecting Multiple Organizations

When aggregating, several options are available for selecting organizations. Consider the following example.



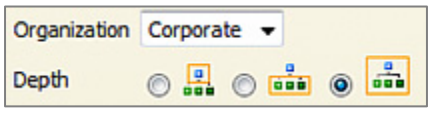
The *This* option uses the *scorecard* navigator to determine the organization.



This enables you to reuse a function (through cut and paste).

If a specific organization is selected, the data from that organization will be used in the calculation, regardless of what *scorecard* the organization is attached to. For example, for all production sites to report on a global *KPI* of overall employee satisfaction, you can aggregate all the data at the Corporate level, including all child organizations, and include that in all production site *scorecards*.

Organization Depth



Options: parent, children, parent and children.

Organization depth controls how deeply the data is aggregated. Selecting parent will only make the calculation from the organization selected in the selection box (or this, if selected). Selecting children sums only the data beneath the selected organization. The final option sums the data of the parent and all children.