



The basis of reporting stems from presentation arcs in role modules. Nothing can show up in a report without a presentation arc relationship. Think of a role module as a drawing canvas. We might have one for the Balance Sheet and another for the Income Statement.

A taxonomy consists of a pallet of elements that can be placed upon our report canvas. We will use some of them, but generally not all of them for any given report. We do this by placing elements into position through the use of presentation arcs. In Snappy Reports this is a simple drag and drop operation.

Calculation arcs are used for mathematical relationships, such as $assets = cash + inventory$. This is again a simple drag and drop operation in Snappy Reports.

Presentation and calculation arcs are specific to the domain of a particular role module (canvas). Each role module will represent a future report.

XBRL is language independent. Our taxonomy should have label translations for any language we wish to render a report in. Labels also have roles. These may include standard, terse or verbose. A standard label may look like *Cash in Bank*. Terse would be a shorted version, such as just *Cash*. Verbose would be the opposite longer version, such as *myAccount Cash in Bank*.

Reports can be rendered in Excel, Word, HTML or as an instance document. Whichever is chosen is immaterial to the data and format of the report. Each should contain the same context, except that instance documents include data for all role modules concurrently, while the other options do not.

Reporting Theory

Reports are essentially a graphical X-Y representation of data. One axis has to be constant and the other variable. Generally, our axis consists of either one or more time periods for a single entity, or multiple entities for a single time period.

For instance, if General Motors is reporting earnings, it may consider just the Buick division over each of the twelve months of 2006, or it may compare multiple divisions (Buick, Pontiac & Chevrolet) over a single time period. It makes little sense to compare the earnings for Buick in January to Pontiac for February. Either we deal with one entity for multiple time periods, or multiple entities for a single time period.

Rendering Theory

An instance document is self contained in that it includes information for all periods and entities related within the document. If we are rendering (drawing) the report for human consumption, we have to pick and choose the information we want.

For instance, taxonomies are language independent. If we have label translations for multiple languages, we can render in only one language. If we have multiple choices in label roles (standard, verbose, terse), we are able to render in only one of those roles.

If we have multiple role modules, we can render in only one of them, and if we have multiple contexts, we must pick either the entity or time period as the constant domain and the remaining as the variable.

We must also choose a general format. Tabular is a cross hatch of multiple columns and multiple rows, similar to an Excel spreadsheet. Linear displays columns in linear descending order. Both display exactly the same data. The only difference is how it appears on the page.

For a simple example, using the Snappy Reports reporting manager displayed below, the taxonomy for Widget Taxonomy has received data only from domain Widget Corporation for the periods of January,

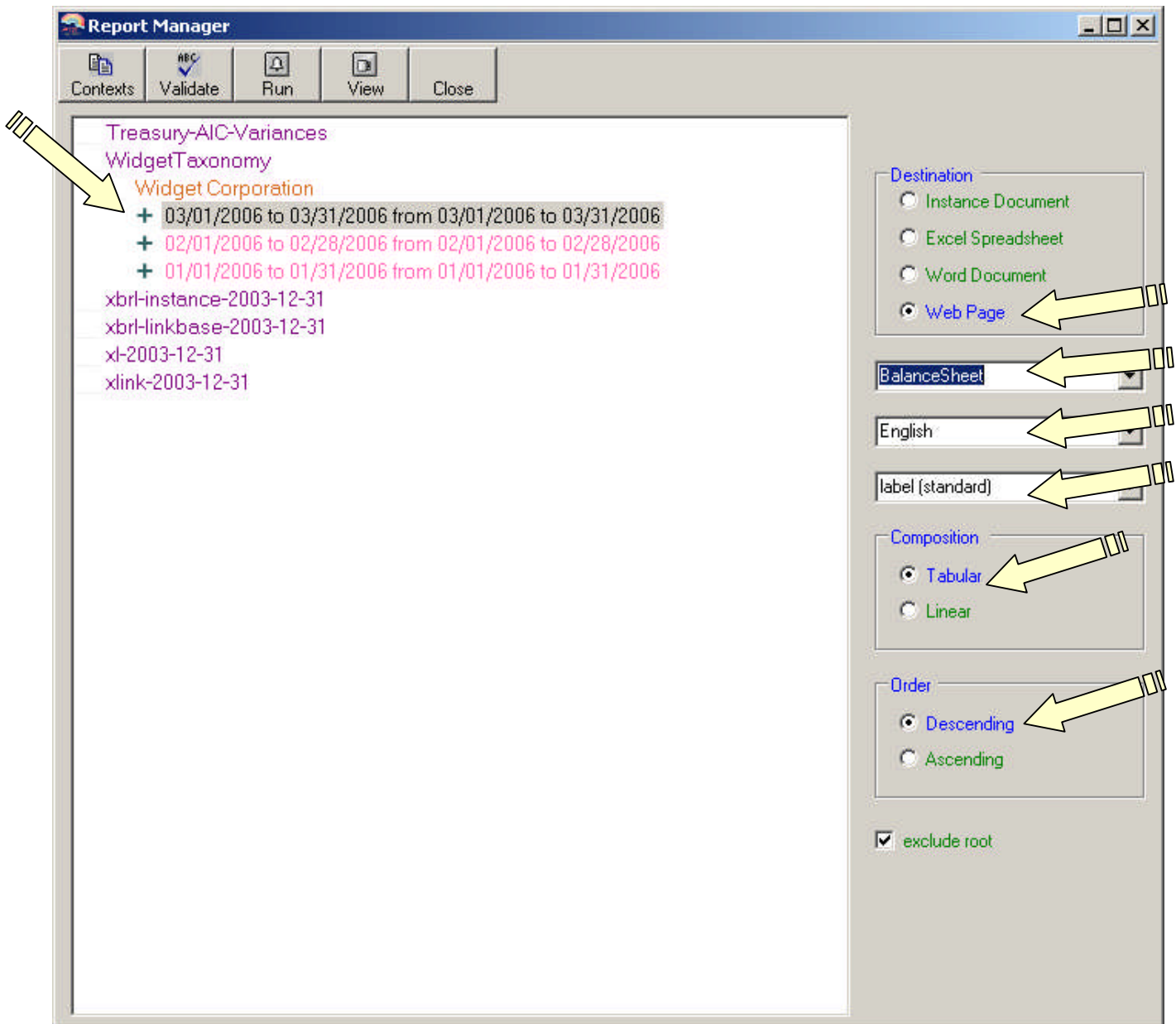


Reporting and Rendering Theory

February and March. We have selected to render all contexts (indicated by the green plus + sign) using the role module for Balance Sheet, in the English language using a standard label role. We have requested that it be displayed in tabular format.

If a variable axis (either time periods or entities) consist of more than one object, we must consider a display order of either ascending or descending. For time periods, that means either January to December or December to January. For entities, it would be alphabetical order.

The view and run options are similar, except that view also displays the report to you, while run only generates it to file. The validation process checks to see if a context is valid, prior to rendering.



Notice that we selected a variable period list for Widget Corporation. Those periods are displayed in descending order (March to January). The report is only a code fragment, since we did not complete construction of the complete balance sheet role module for Widget for our sample.



Rendering Disclosures

Disclosure is an important facet to reporting. Elements within a taxonomy do not have to be numerical. They can be text.

Context Aligned Disclosures

There are circumstances where a report requires numerical information from the remainder of the report as part of a textual explanation. For instance the statement, *Ending inventory of \$1380 has been marked down to the lower of cost or value*, includes the value \$1380 which is derived from the current value of an element that is outside the scope of the disclosure.

The numerical value will change depending on context. This makes things a bit more complicated. For instance, if we are preparing a tabular report for both January and February, the ending inventory balance would be different for each.

Snappy Reports handles this through a formatted reference in the disclosure. It looks like this.

Inventory has been marked down to {A300:#,###,##0} based on lower of cost or market.

The symbol A300 refers to a taxonomy element having an identifier of A300, in this case it is the element for inventory. The #,###,##0 is the format string for how the number is to look when printed. Snappy Reports uses standard Microsoft formatting rules.

Placement of the disclosure depends entirely where you place the disclosing element within the role module through the use of presentation arcs. A disclosure must be placed using a presentation arc, or it will not appear when the report is rendered. A context aligned disclosure aligns similar with numerical / monetary items on the report. If the report is tabular, these disclosures will repeat once for each context within the frame designed for any other data within the context.



BalanceSheet Widget Corporation

	from 03/01/2006 to 03/31/2006	from 02/01/2006 to 02/28/2006	from 01/01/2006 to 01/31/2006
Assets			
Current Assets			
Cash In Bank	12,633.00	12,633.00	1,263.30
Cash- Bank Of New Hampshire	4,295.70	4,295.70	429.57
Cash- Bank Of New Jersey	3,416.10	3,416.10	341.61
Cash- Bank Of New York	4,921.20	4,921.20	492.12
Accounts Receivable	34,786.50	34,786.50	3,478.65
Inventory	13,800.00	13,800.00	1,380.00
	Inventory has been marked down to 13,800 based on lower of cost or market.	Inventory has been marked down to 13,800 based on lower of cost or market.	Inventory has been marked down to 1,380 based on lower of cost or market.
Disclose Inventory			
Total Current Assets	61,219.50	61,219.50	6,121.95

Label Aligned Disclosures

We may want our disclosures to print underneath the label for the element, to which the disclosure pertains. We cannot do this with a context disclosure, because disclosures vary depending on context.

We can hard code a disclosure directly into the taxonomy as a label to an element. The down side it that disclosure text is permanent and cannot be changed based on data or circumstance.

Scenario Aligned Disclosures

Scenarios by definition add additional information to the taxonomy. Scenarios can be used to place disclosures under element labels and allowing them to change based on data and circumstance. The use of scenarios for disclosures eliminates the pitfalls of static label aligned disclosures discussed previously.

Segment Aligned Disclosures

Segments are similar to scenarios, except that they describe an entity. We previously discussed that a report is an X-Y rendition of entities v time periods. If the entity is a constant and time periods are variable, we are looking good. We can place the segment label in the label column of the report. However, if the report displays multiple entities over a single period of time, we are no longer able to do that. Unfortunately, we have no way of determining how a user will select parameters for a report. Segment aligned disclosures can work similar to scenario disclosures only if the entity is a constant.



Neither segment nor scenario aligned disclosures can use the formatted foreign reference option {A300:#,###,##0}, since foreign references are context related.

Rendering Spreadsheets

Rendering via an Excel spreadsheet is similar to other forms of rendering, except for the fact that calculation arcs require special handling. Spreadsheets are formula based. This means where calculation arcs produce a total, the spreadsheet should produce the formula for the total, rather than the numerical value of the result. For example, if *cash + inventory = current assets* then the cell for current assets would equal something like $=sum(B10+B11)$, instead of an actual value. Placing resulting values in cells may cause problems should someone modify a rendered spreadsheet without realization that calculated totals are not being recalculated, because the rendering was not formula based.

Rendering to the Web

Although XBRL is designed to handle data, there is little in the specification that deals with rendering formatting. By formatting, I'm referring to fonts, colors, headings and logos.

These can be handled through the use of cascading style sheets, which are developed outside the scope of XBRL. Snappy Reports folds in style sheets into web based reports, when available in order to provide the look and feel desired.