Mobile Analytics Kit Code Examples: iOS

Sybase Unwired Platform 2.2

Document ID: DC01681-01-0220-01

Last Revised: August 2012

Copyright © 2012 by Sybase, Inc. All rights reserved.

This publication pertains to Sybase software and to any subsequent release until otherwise indicated in new editions or technical notes. Information in this document is subject to change without notice. The software described herein is furnished under a license agreement, and it may be used or copied only in accordance with the terms of that agreement.

Upgrades are provided only at regularly scheduled software release dates. No part of this publication may be reproduced, transmitted, or translated in any form or by any means, electronic, mechanical, manual, optical, or otherwise, without the prior written permission of Sybase, Inc.

Sybase trademarks can be viewed at the Sybase trademarks page at http://www.sybase.com/detail?id=1011207. Sybase and the marks listed are trademarks of Sybase, Inc. ® indicates registration in the United States of America.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world.

Java and all Java-based marks are trademarks or registered trademarks of Oracle and/or its affiliates in the U.S. and other countries.

Unicode and the Unicode Logo are registered trademarks of Unicode, Inc.

IBM and Tivoli are registered trademarks of International Business Machines Corporation in the United States, other countries, or both.

All other company and product names mentioned may be trademarks of the respective companies with which they are associated.

Use, duplication, or disclosure by the government is subject to the restrictions set forth in subparagraph (c)(1)(ii) of DFARS 52.227-7013 for the DOD and as set forth in FAR 52.227-19(a)-(d) for civilian agencies.

Sybase, Inc., One Sybase Drive, Dublin, CA 94568.

Contents

Introduction 4

Code Examples Setup 4

Code Examples 5

Basic Dashboards 5

Examples of Chart Configuration 6

Example of Dashboard Layout 6

Dynamic Analytic Panel 8

Example of Variable Definition 8

DAPanel Configuration in Dashboard and DashboardLayout 9

Drilldown Charts 12

Example of Drilldown Configuration 13

Dashboard Filters 17

Example of Filter Definition/Chart Filter Configuration 17

Zooming Charts 20

Example of Zooming Chart Configuration 21

What-If Analysis 23

Example of What-If Analysis Configuration 24

MAAnalyticChartView 28

Example of AnalyticChartView Configuration 28

Influence Diagram 29

Example of Influence Diagram Configuration 29

Predictive Analysis and Threshold Reference 33

Example of Threshold and Predictive Analysis Configuration 33

MAKit2.2 New Features 36

Examples of New Chart Configuration 37

Time Series Drawing 38

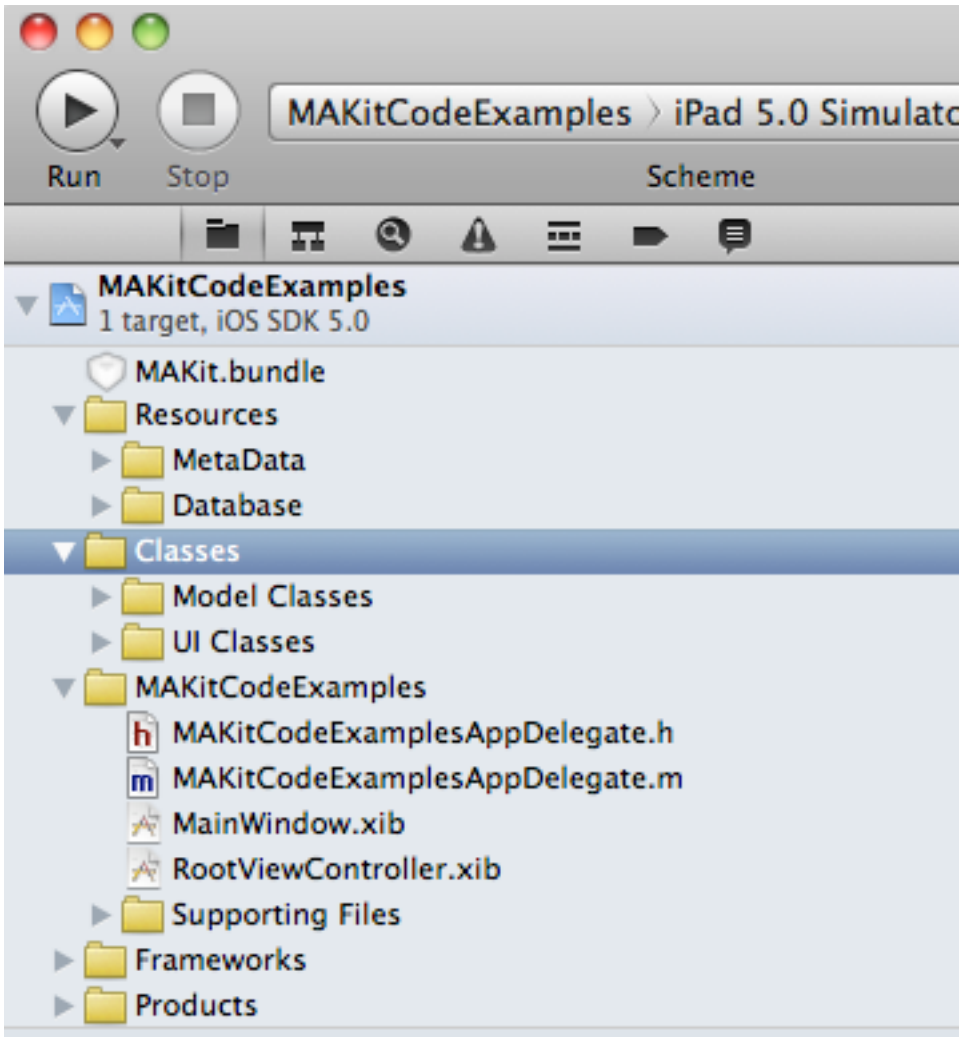
Introduction

The Mobile Analytics Kit (MAKit) code examples demonstrate the core features of MAKit charts, such as dashboard configuration, applying analytical capability to the charts using the dynamic analytic panels, chart drilldown and filtering, and semantic zooming. Additionally, all charts have a built-in e-mail sharing feature, which captures a snapshot image of the chart, and allows the device user to first annotate, then send the image via the configured e-mail application. MAKit can be used with the Sybase® Unwired Platform Hybrid Web Container.

Code Examples Setup

Note: These setup instructions are specific to the Xcode 4.2 IDE. For earlier versions of Xcode, the steps may differ.

1. Copy `MAKitCodeExamples.zip` to your Mac.
2. Unzip `MAKitCodeExamples.zip`. This creates a folder called `MAKitCodeExamples`.
3. In the Xcode IDE, double-click **MAKitCodeExamples.xcodeproj**. You should see a screen similar to:



- `Classes` – contains the application source code for the code examples.
- `XIB Files` – contains the Interface Builder files used in the code examples.
- `Database` – contains the SQLite® database `MAKitSampleDb.sqlite`, which is the datasource for the code examples.
- `MetaData` – has all the metadata files for each individual feature in the code examples.

All other folders are created by default for any Xcode 4.2 project.

4. To run the code example in iPad Simulator, select **Product > Run**, or click the **Run** toolbar button.

To run the code example on an Apple iPad, install a valid device-provisioning profile in the Xcode IDE before deploying to the device.

Code Examples

Basic Dashboards

This section demonstrates the chart types supported by MAKit. You can group related charts together to create a meaningful dashboard, for example, a sales or stock information dashboard. See the `Dashboard_ChartTypes.xml` metadata file content to learn how to configure different chart types.

Configure all charts using the *<Dashboard>…</Dashboard>* tags.

Examples of Chart Configuration

```
<Chart Name="ColumnChart" Title="Region/Product Sales" ChartType="Column" Query="SalesOrder">
    <Category Format="" Column="Region" DisplayName="Region" />
    <Series Format="" Column="Product" DisplayName="Region" />
    <Values>
         <Value Format="Currency" Expression="SalesAmount" ValueSequence="1"
                DisplayName="Sales Amount" /> </Values>
</Chart>

<Chart Name="BarChart" Title="Salary By Dept/Gender" ChartType="Bar" Query="DeptSalary">
    <Category Format="" Column="Department" DisplayName="Department" />
    <Series Format="" Column="Sex" DisplayName="Gender" />
    <Values>
        <Value Format="Rounded2" Expression="Salary" ValueSequence="1" DisplayName="Salary" />
    </Values>
 </Chart>

<Chart Name="LineChart" Title="Number Of Sales Orders" ChartType="Line" Query="NumberOfOrders">
    <Category Format="" Column="Year" DisplayName="Year" />
    <Series Format="" Column="Region" DisplayName="Region" />
    <Values>
        <Value Format="Rounded0" Expression="OrderCount" ValueSequence="1"
               DisplayName="Number Of Orders" />
    </Values>
</Chart>

<Chart Name="PieChart" Title="Printer Sales" ChartType="Pie" Query="PrinterSales">
    <Category Format="" Column="SalesRep" DisplayName="Sales Rep" />
    <Series Format="" Column="Product" DisplayName="Type" />
    <Values>
        <Value Format="Currency" Expression="Sales" ValueSequence="1" DisplayName="Sales Amount " />
    </Values>
</Chart>

<Chart Name="BubbleChart" Title="US Crime Data" ChartType="Bubble" Query="CrimeData">
    <Category Format="" Column="State" DisplayName="State" />
    <Series Format="" Column="CrimeType" DisplayName="CrimeType" />
    <Values>
        <Value Format="Rounded2" Expression="Ratio" ValueSequence="1" DisplayName="Crime Ratio" />
        <Value Format="Rounded2" Expression="Ratio" ValueSequence="2" DisplayName="Crime Ratio" />
    </Values>
</Chart>
```

The metadata file also configures the dashboard layouts for either portrait or landscape mode.

Example of Dashboard Layout

```
<DashboardLayout>
       <Landscape>
            <Layout Name="ColumnChart" Top="1" Left="1" Width="49" Height="32" />
            <Layout Name="BarChart" Top="1" Left="51" Width="48" Height="32" />
            <Layout Name="LineChart" Top="34" Left="1" Width="49" Height="32" />
            <Layout Name="PieChart" Top="34" Left="51" Width="48" Height="32" />
            <Layout Name="BubbleChart" Top="67" Left="1" Width="98" Height="32" />
        </Landscape>
        <Portrait>
            <Layout Name="ColumnChart" Top="1" Left="1" Width="49" Height="32" />
            <Layout Name="BarChart" Top="1" Left="51" Width="48" Height="32" />
            <Layout Name="LineChart" Top="34" Left="1" Width="49" Height="32" />
            <Layout Name="PieChart" Top="34" Left="51" Width="48" Height="32" />
            <Layout Name="BubbleChart" Top="67" Left="1" Width="98" Height="32" />
        </Portrait>
  </DashboardLayout>
```

This dashboard layout configuration renders a chart similar to:

Dynamic Analytic Panel

This section demonstrates the dynamic analytic (DA) capabilities of MAKit. Dynamic analytics is a powerful feature that lets you interactively visualize reports by changing some parameters at runtime, and refreshing the charts based on the new values.

The heart of dynamic analytics is the variable definition in the metadata, which constructs the value axis expressions using the variables. When the device user changes a variable value at runtime, MAKit evaluates the value axis expression, and refreshes the charts based on the new value.

Define variables in the metadata using the *<Variables>……</Variables>* tags, and group them in the Dynamic Analytic panel metadata using the *<DAPanel>……</DAPanel>* tags. Configure the Dynamic Analytic panel using a dashboard layout that is similar to that used for configuring charts.

MAKit automatically generates UI controls to manipulate the values of these variables:

Variable Type	User Interface Generated by MAKit
integer	Slider control
float	PickerView control
boolean	Switch control
enum	PickerView control
datetime	Date Picker control

Example of Variable Definition

```
<Variables>

   <Variable Name="$salesdrive" DisplayName="Sales Promotion ?" VariableType="boolean" Value="0" />

   <Variable Name="$region" DisplayName="Sales Promotion in " VariableType="enum" Selection="0">
      <Choices>
          <Choice Display="Eastern" Value="Eastern" />
          <Choice Display="Western" Value="Western" />
          <Choice Display="Central" Value="Central" />
          <Choice Display="South" Value="South" />
          <Choice Display="Canada" Value="Canada" />
       </Choices>
  </Variable>
  <Variable Name="$dept" DisplayName="Pay Raise Department " VariableType="enum" Selection="0">
     <Choices>
         <Choice Display="R &amp; D" Value="R &amp; D" />
         <Choice Display="Sales" Value="Sales" />
         <Choice Display="Finance" Value="Finance" />
         <Choice Display="Marketing" Value="Marketing" />
         <Choice Display="Shipping" Value="Shipping" />
     </Choices>
   </Variable>
   <Variable Name="$hike" DisplayName="Pay Raise % " VariableType="float" Value="0" Min="0"
             Max="18" Decimals="1" />
</Variables>
```

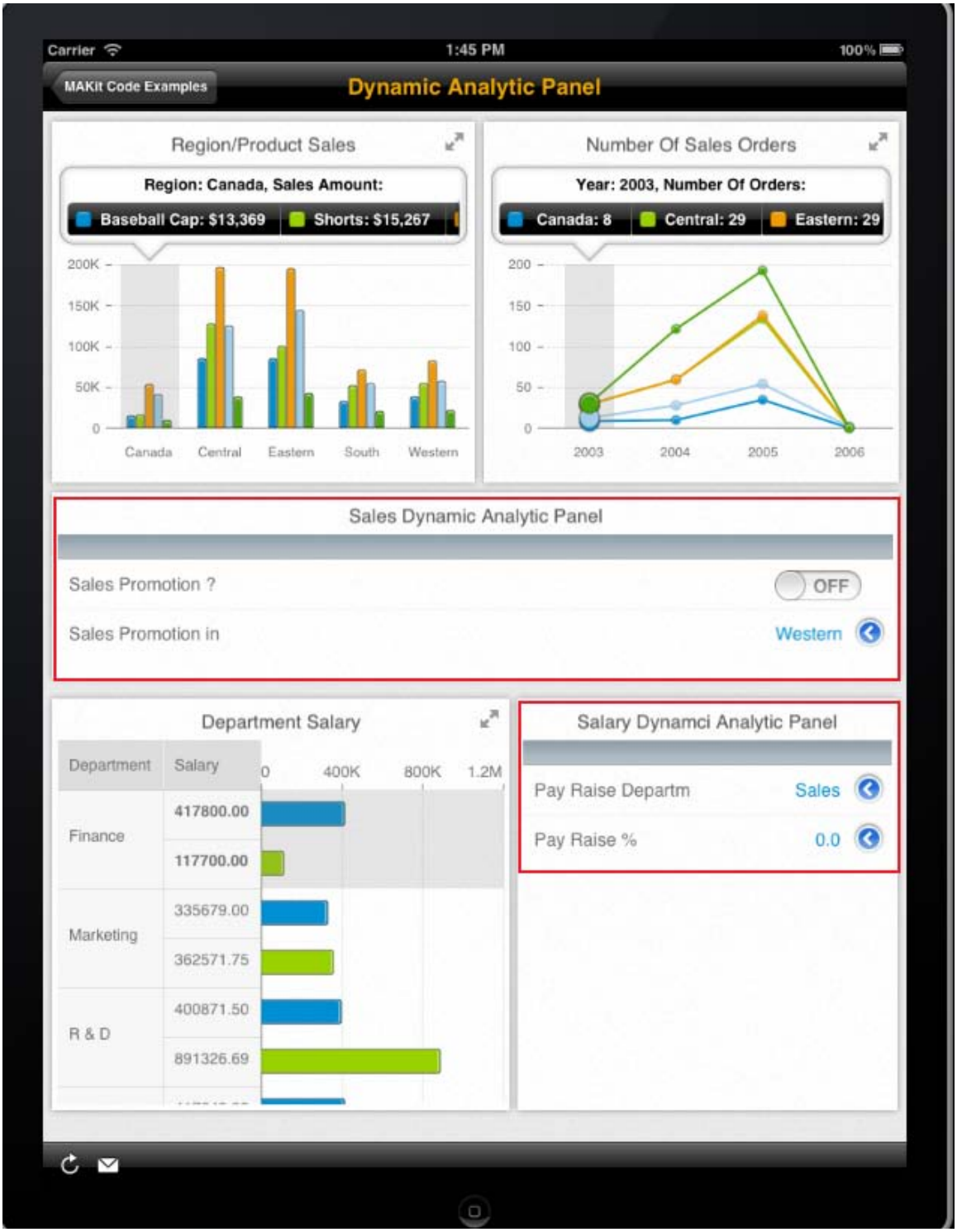
The variables defined in the metadata are used in value expressions of chart configuration as shown here:

```
<Chart Name="ProductSales" Title="Region/Product Sales" ChartType="Column" Query="SalesOrder">
   <Category Format="" Column="Region" DisplayName="Region" />
   <Series Format="" Column="Product" DisplayName="Product" />
   <Values>
       <Value Format="Currency" Expression="if($salesdrive==1,SalesAmount+(SalesAmount *
                            0.30),if(Region==$region, SalesAmount+(SalesAmount * 0.80),SalesAmount))"
                    ValueSequence="1" DisplayName="Sales Amount" />
   </Values>
 </Chart>

<Chart Name="Orders" Title="Number Of Sales Orders" ChartType="Line" Query="NumberOfOrders">
   <Category Format="" Column="Year" DisplayName="Year" />
   <Series Format="" Column="Region" DisplayName="Region" />
   <Values>
      <Value Format="Rounded0" Expression="if($salesdrive==1,OrderCount * 2,
             if(Region==$region, OrderCount * 6,OrderCount))" ValueSequence="1"
                      DisplayName="Number Of Orders" />
      </Values>
</Chart>

<Chart Name="PayRaise" Title="Department Salary" ChartType="Bar" Query="DeptSalary">
    <Category Format="" Column="Department" DisplayName="Department" />
    <Series Format="" Column="Sex" DisplayName="Gender" />
    <Values>
       <Value Format="Rounded2" Expression="if(Department==$dept,(Salary + Salary *
                                ($hike/100)),Salary)" ValueSequence="1" DisplayName="Salary" />
    </Values>
</Chart>
```

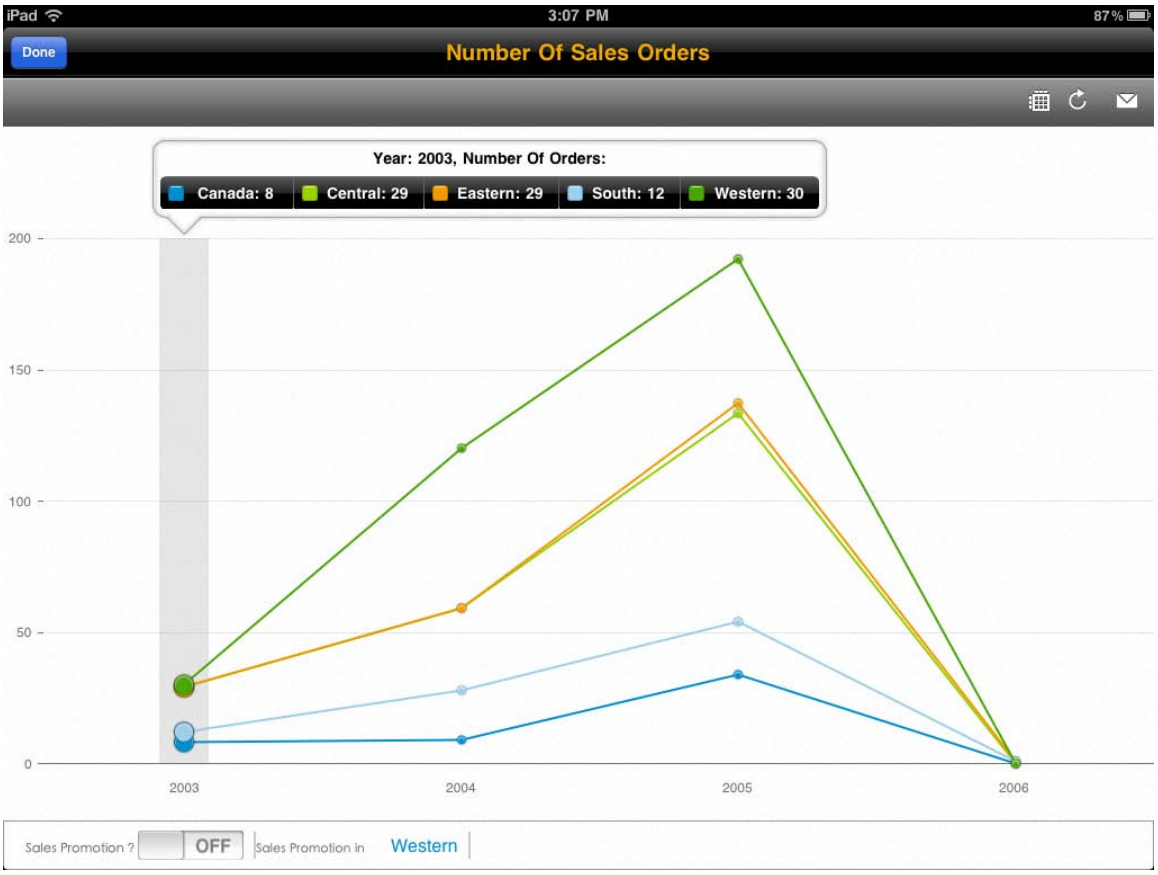
Dynamic Analytic Panel Configuration in Dashboard and DashboardLayout

```
<DAPanel Name="SalesDA" Title="Sales Dynamic Analytic Panel">
    <VariableName Name="$salesdrive" Sequence="1" />
    <VariableName Name="$region" Sequence="2" />
</DAPanel>

<DAPanel Name="EmpDA" Title="Salary Dynamic Analytic Panel">
    <VariableName Name="$dept" Sequence="1" />
    <VariableName Name="$hike" Sequence="2" />
</DAPanel>
<DashboardLayout>
    <Landscape>
        <Layout Name="ProductSales" Top="1" Left="1" Width="49" Height="35" />
        <Layout Name="Orders" Top="1" Left="51" Width="48" Height="35" />
        <Layout Name="SalesDA" Top="37" Left="1" Width="98" Height="20" />
        <Layout Name="PayRaise" Top="60" Left="1" Width="53" Height="35" />
        <Layout Name="EmpDA" Top="60" Left="55" Width="44" Height="35" />
    </Landscape>

    <Portrait>
        <Layout Name="ProductSales" Top="1" Left="1" Width="49" Height="35" />
        <Layout Name="Orders" Top="1" Left="51" Width="48" Height="35" />
        <Layout Name="SalesDA" Top="37" Left="1" Width="98" Height="18" />
        <Layout Name="PayRaise" Top="57" Left="1" Width="53" Height="40" />
        <Layout Name="EmpDA" Top="57" Left="55" Width="44" Height="40" />
    </Portrait>
</DashboardLayout>
```

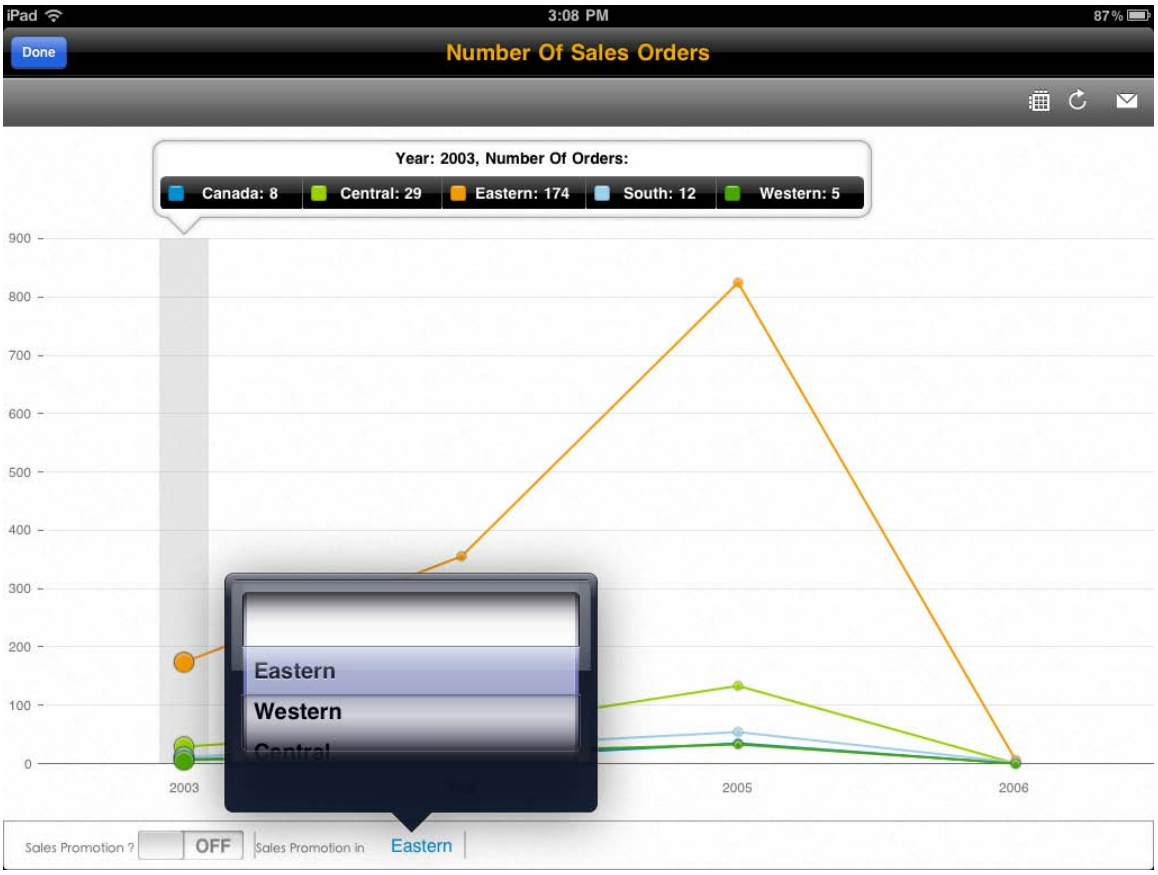
The Dynamic Analytic panel renders in the dashboard in the same manner as individual chart items. When variable values are changed, relevant charts are refreshed based on the expression evaluation.



The Dynamic Analytic panel appears when the chart is in full-view mode:



This figure shows the refreshed chart based on a new variable value ("Eastern"):



Drilldown Charts

MAKit lets you drill down by category or by series, or both, on a single chart. You can define any level of drilldown chart in the metadata definition.

You can drill down only when the chart is in full-view mode, by double-tapping the chart (drilldown by category) or by double-tapping either the category value or the series value in the data grid table of the chart.

Configure the drilldown charts using `<DrillDownGroup>….</DrillDownGroup>` tags; configure individual charts at each level using `<Chart> …. </Chart>` tags.

See the `Drilldown_Sales.xml` and `Drilldown_Energy.xml` metadata files in the code example to see drilldown configurations.

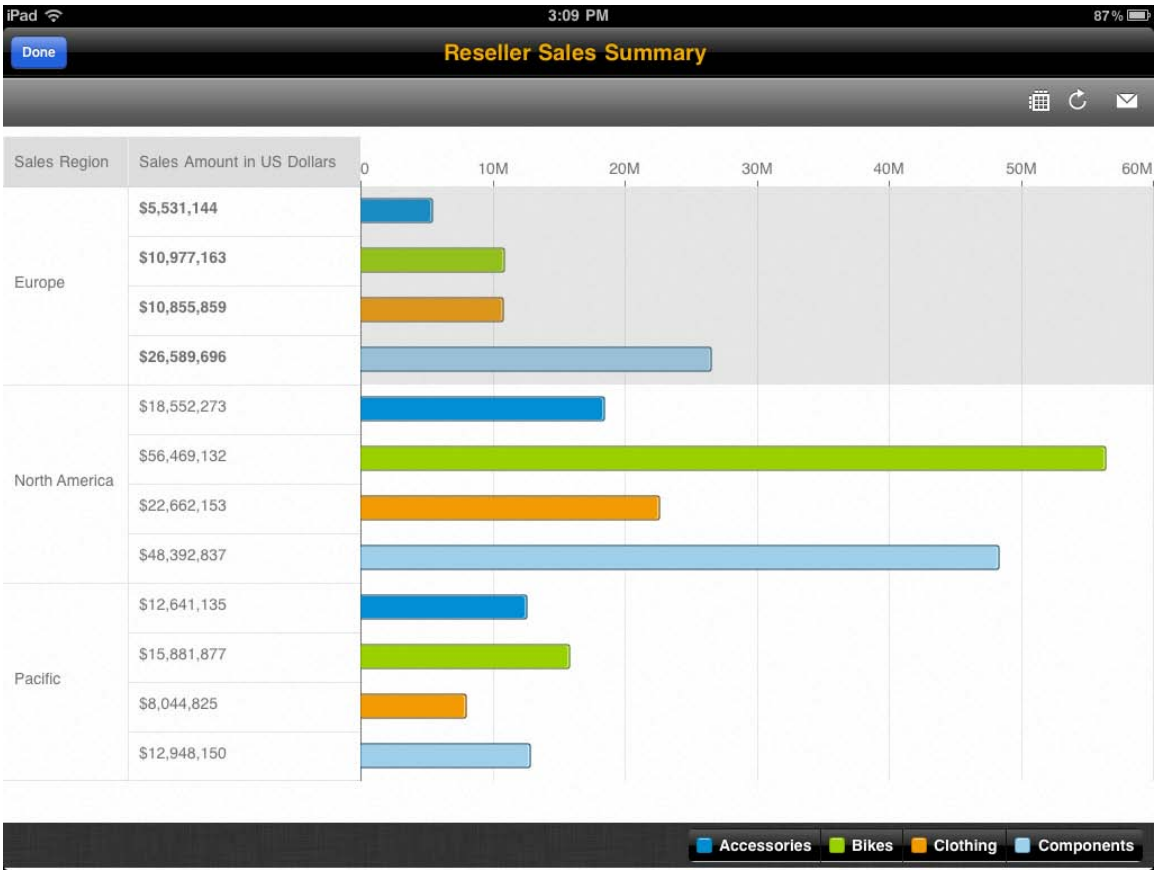
Example of Drilldown Configuration

```
<DrillDownGroup Name="Reseller" Title="Reseller Drilldown Charts">
   <Chart Name="ResellerSales" Title="Reseller Sales Summary " ChartType="Bar" ChartSequence="1"
                                    Query="ResellerSummary" >
       <Category Format="" Column="Region" DisplayName="Sales Region" />
       <Series Format="" Column="ProductCategory" DisplayName="ProductCategory" />
       <Values>
         <Value Format="Currency" Expression="SalesAmount" ValueSequence="1"
                                 DisplayName="Sales Amount in US Dollars" />
        </Values>
    </Chart>
    <Chart Name="SalesByCountry" Title="Reseller Sales By Country" ChartType="Bar" ChartSequence="2"
                                      Query="ResellerByCountry" >
        <Category Format="" Column="Country" DisplayName="Country" />
        <Series Format="" Column="ProductCategory" DisplayName="Product Category" />
        <Values>
             <Value Format="Currency" Expression="SalesAmount" ValueSequence="1"
                                    DisplayName="Sales Amount in US Dollars" />
        </Values>
    </Chart>
    <Chart Name="SalesByCategory" Title="Sales By Product Category " ChartType="Bar" ChartSequence="3"
                                       Query="ResellerByCategory">
        <Category Format="" Column="ProductSubCategory" DisplayName="Product Sub Category" />
        <Series Format="" Column="Country" DisplayName="Country" />
        <Values>
            <Value Format="Currency" Expression="SalesAmount" ValueSequence="1"
                                   DisplayName="Sales Amount in US Dollars" />
         </Values>
     </Chart>
 </DrillDownGroup>
```

The Drilldown chart renders similarly to other charts configured in the dashboard layout:

Carrier
1:52 PM
100%
MAKit Code Examples
Drilldown Example 1
Reseller Sales Summary
Sales Region
Sales Amount in US Dollars
0
10M
20M
30M
40M
50M
60M
Europe
$5,531,144
$10,977,163
$10,855,859
$26,589,696
North America
$18,552,273
$56,469,132
$22,662,153
$48,392,837
Online Sales Drive
Driver For Sales: Manufacturer, Sales Amount in US Dollars:
Europe: $1,207,251
North America: $2,767,663
Pacific: $2,023,208
5M
4M
3M
2M
1M
0
Manufacturer
Other
Quality
Television
Advertisement

Drilldown chart in full-view mode:



Drilldown by category (double-tap “Europe” in the bar chart):

Drilldown by series (double-tap ”Bikes” in the table*)*:

Dashboard Filters

MAKit dashboard filters let you filter chart data using query arguments. Filters are global in nature and affect all charts associated with that filter. Data that does not fulfill the range or criteria specified is filtered out.

Filters are initially defined in the metadata using *<Dashboard> …. </Dashboard>* tags, and individual charts configured in the metadata can apply previously defined filters.

There are two types of filter: a range filter lets you specify an integer range, and a picklist filter lets you select discrete sets of data criteria.

Example of Filter Definition/Chart Filter Configuration

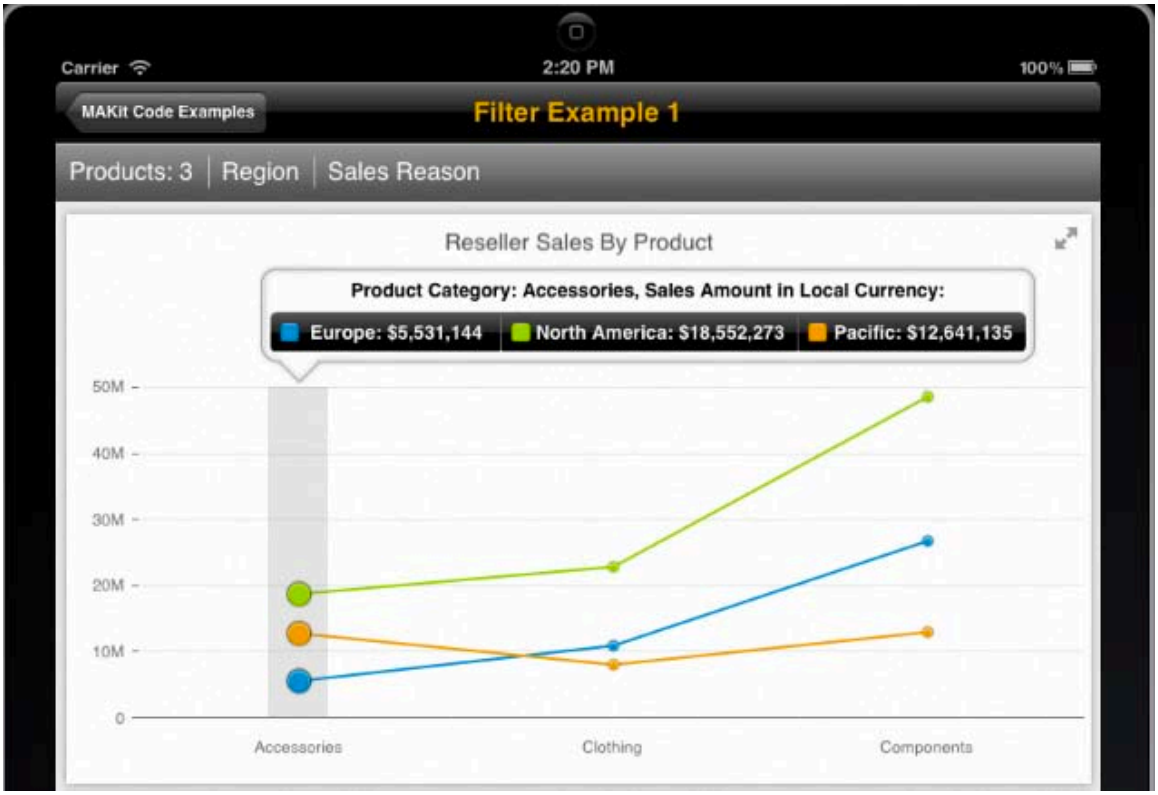
```
<!-- Filter Definition in Dashboard -->
<Filters>
  <Filter Name="Products" Sequence="1" SourceQuery="ResellerQry" FilterType="picklist"
              Column="ProductCategory" />
  <Filter Name="Region" Sequence="2" SourceQuery="ResellerQry" FilterType="picklist" Column="Region" />
  <Filter Name="Sales Reason" Sequence="3" SourceQuery="OnlineSalesQry" FilterType="picklist"
              Column="SalesReason" /
</Filters>


<!-- Configuring Charts with Filters -->

<Chart Name="ResellerSales" Title="Reseller Sales By Product" ChartType="Column" Query="ResellerQry">
    <Category Format="" Column="ProductCategory" DisplayName="Product Category" />
    <Series Format="" Column="Region" DisplayName="Sales Region"/>
    <Values>
       <Value Format="Currency" Expression="SalesAmount" ValueSequence="1" DisplayName=" " />
    </Values>
    <Filters>
        <FilterName Name="Products" />
        <FilterName Name="Region" />
    </Filters>
</Chart>

<Chart Name="OnlineSales" Title="Online Sales Drive" ChartType="StackedColumn" Query="OnlineSalesQry">
   <Category Format="" Column="SalesReason" DisplayName="Sales Reason" />
   <Series Format="" Column="Region" DisplayName="Sales Region"/>
   <Values>
      <Value Format="Currency" Expression="SalesAmount" ValueSequence="1" DisplayName="" />
   </Values>
   <Filters>
       <FilterName Name="Sales Reason" />
       <FilterName Name="Region" />
   </Filters>
</Chart>
```

Dashboard filters work only in the dashboard view; they are unavailable in full-view mode.

The filters defined on the dashboard appear in the top row:

Picklist filter:

Result of filters applied*:*



Zooming Charts

Semantic zooming lets device users view a large data set in different scales—for example, an overview of a large chart by zooming out, or every detail by zooming in.

Configure zooming charts, and the behavior of a chart at each zooming level, in the metadata configuration file.

Like drilldown charts, zooming can be performed only in full-view mode, via hierarchy buttons.

Example of Zooming Chart Configuration

```
<Chart Name="ZoomingChart" Title="Sales By Product Category" ChartType="Column">
   <Category Format="" DisplayName="" />
   <Values>
     <Value Format="Rounded2" Expression="SalesAmount" ValueSequence="1" DisplayName="Sales Amount" />
    </Values>
    <ZoomingLevels>
         <ZoomingLevel Query="SalesByCategory" CategoryColumn="ProductCategory"
                       Title="ProductCategory" />
         <ZoomingLevel Query="SalesBySubCategory" CategoryColumn="SubCategory"
                       Title="ProductSubCategory" />
         <ZoomingLevel Query="SalesBySubCategoryAndYear" CategoryColumn="Year" Title="Year" />
     </ZoomingLevels>
</Chart>
```

Chart at ZoomLevel=1:

Chart at ZoomLevel=2:

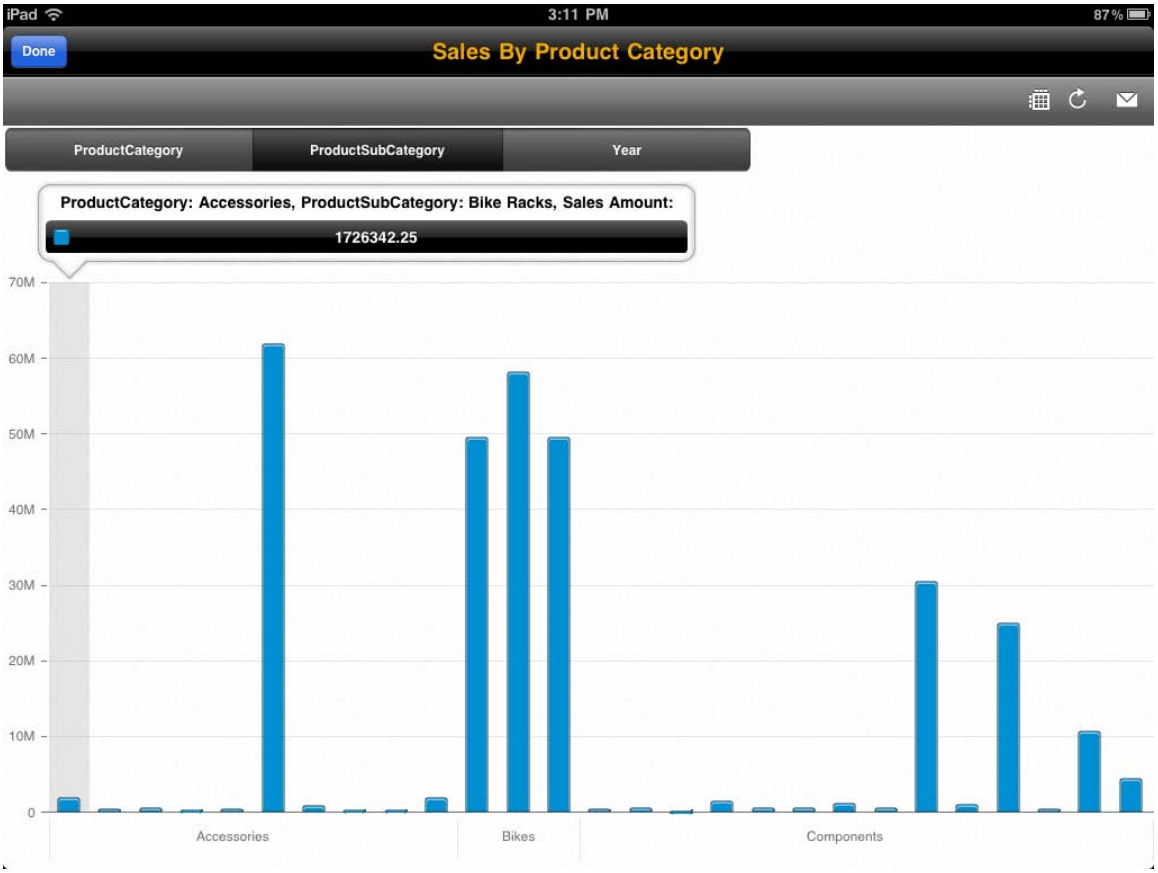
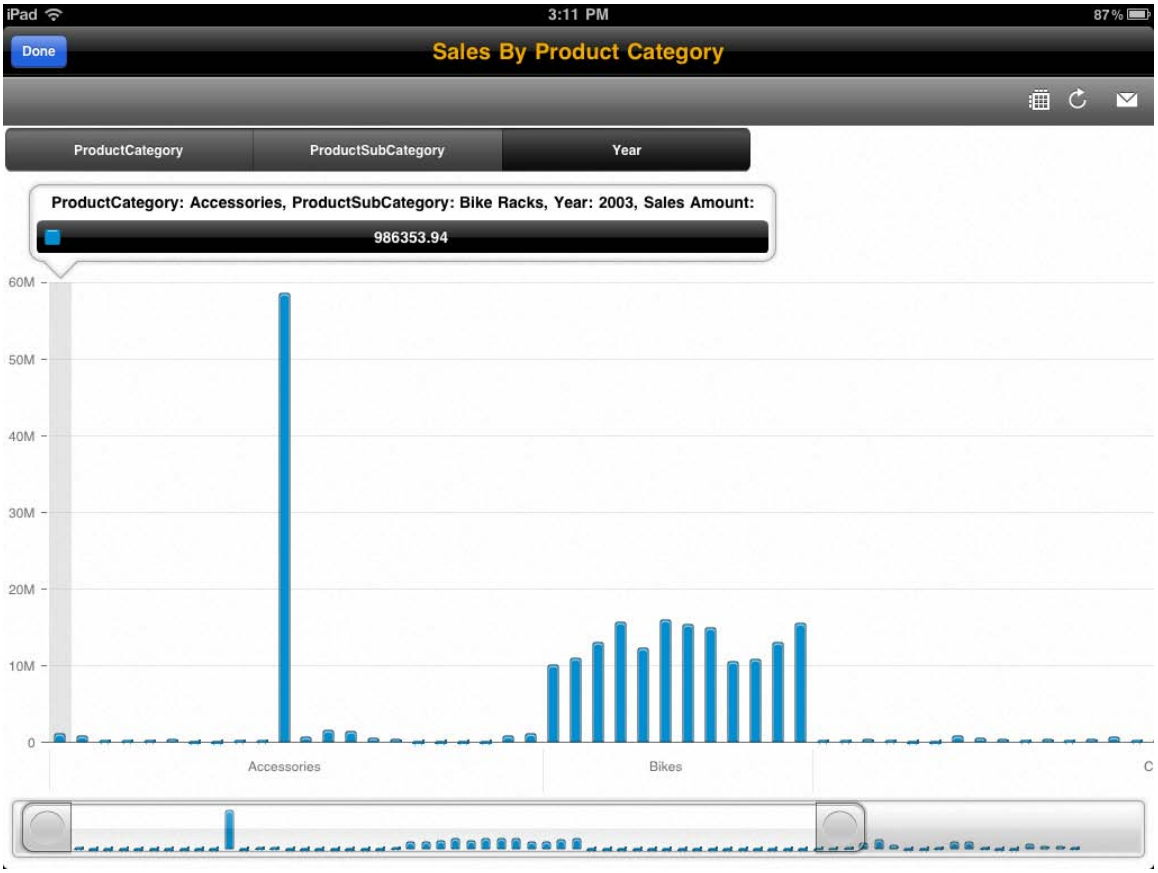


Chart at ZoomLevel=3:



What-If Analysis

What-if analysis is similar to dynamic analytics. Both allow the device user to change data behind the chart and immediately see the chart updated; both map the data to UI controls on the screen. However, while dynamic analytics changes the variables defined in expressions, what-if analysis changes data retrieved from the datasource.

Configure what-if analysis in the metadata, using what-if columns, which are used in value expressions in the chart configuration.

Configure what-if columns using `<WhatIfColumns>…</WhatIfColumns>` tags.

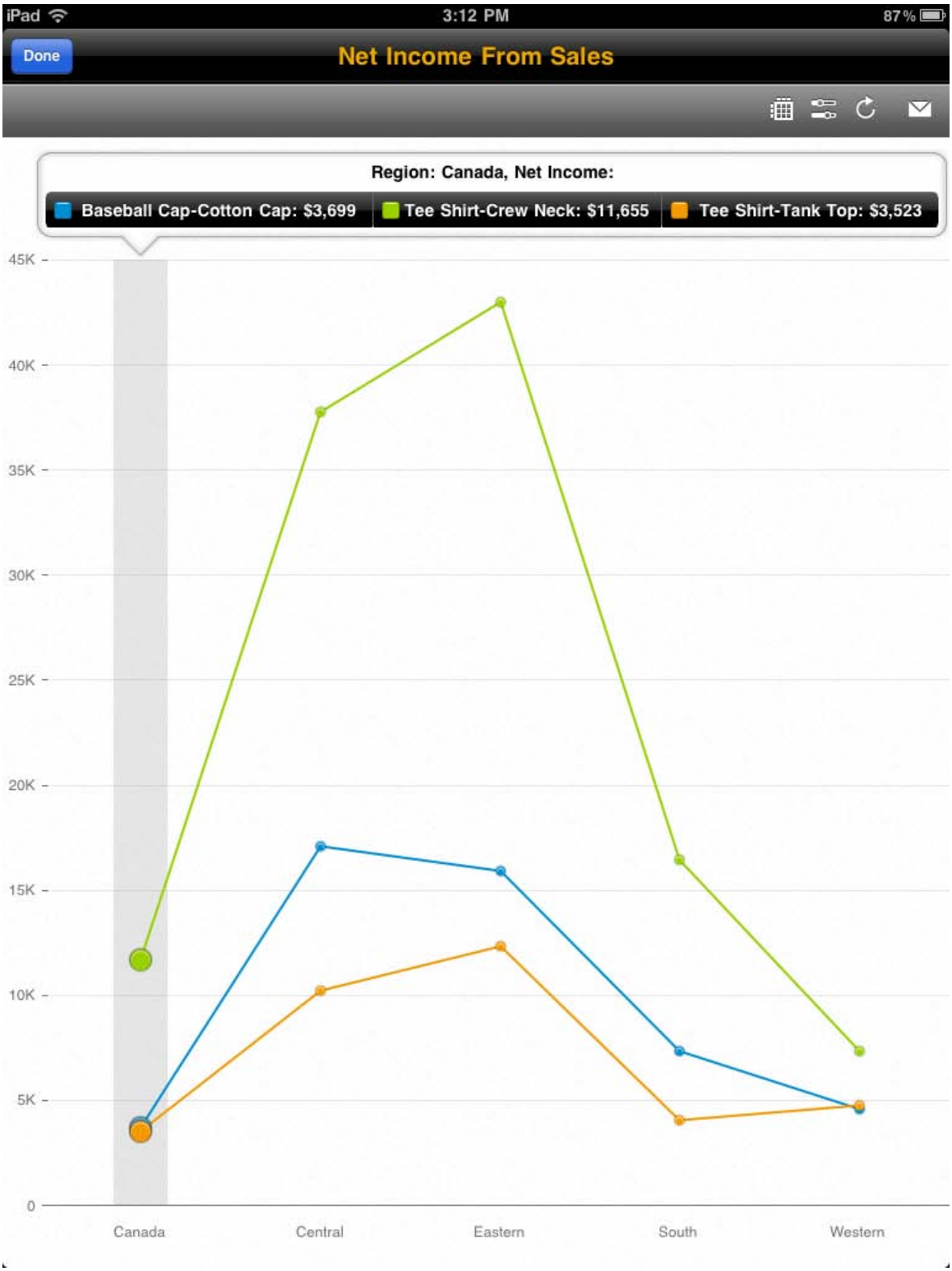
Example of What-If Analysis Configuration

```
<Chart Name="WhatIfChart" Title="Net Income From Sales" ChartType="Line" Query="ProductSales">
   <Category Format="" Column="Region" DisplayName="Region" />
   <Series Format="" Column="Product" DisplayName="Product" />
   <Values>
       <Value Format="Currency" Expression="((UnitPrice * Quantity) - ((Cost_Of_Revenue * Quantity) +
                           (RD_Cost * Quantity) + (Mkt_Cost * Quantity) + (Admin_Cost * Quantity)))"
                          ValueSequence="1" DisplayName="Net Income" />
  </Values>
  <!-- What If Columns -->
  <whatIfColumns>
     <Column Name="UnitPrice" DisplayName="Unit Price" VariableType="float" Min="1.0" Max="110.0"
                             Step="1.0" Decimals="2" />
     <Column Name="Cost_Of_Revenue" DisplayName="Cost Of Revenue" VariableType="integer" Min="1"
                           Max="40" />
     <Column Name="RD_Cost" DisplayName="R &amp; D Cost" VariableType="integer" Min="1" Max="18"  />
     <Column Name="Mkt_Cost" DisplayName="Marketing Cost" VariableType="float" Min="1.0" Max="10.0"
                             Step="1.0" Decimals="2" />
  </whatIfColumns>
</Chart>
```

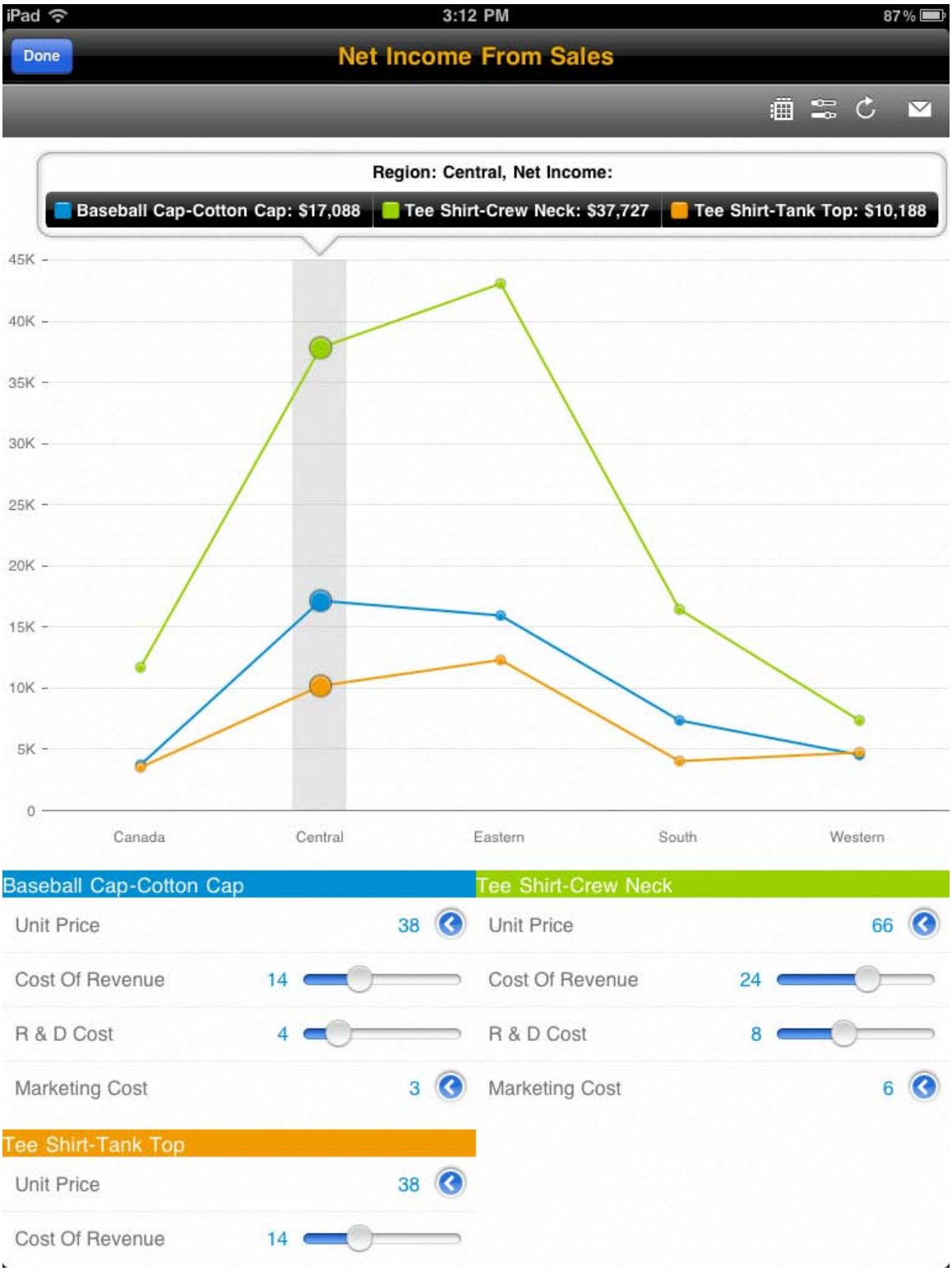
What-if analysis works only in full-view mode, rendering a UI control for each what-if column in the metadata.

To see what-if analysis behavior, select a category and tap the **What-If** button in the toolbar. Tap the button again to close the What-If panel. When the What-If panel is open, changing the values in a what-if column immediately updates the chart for the selected category.

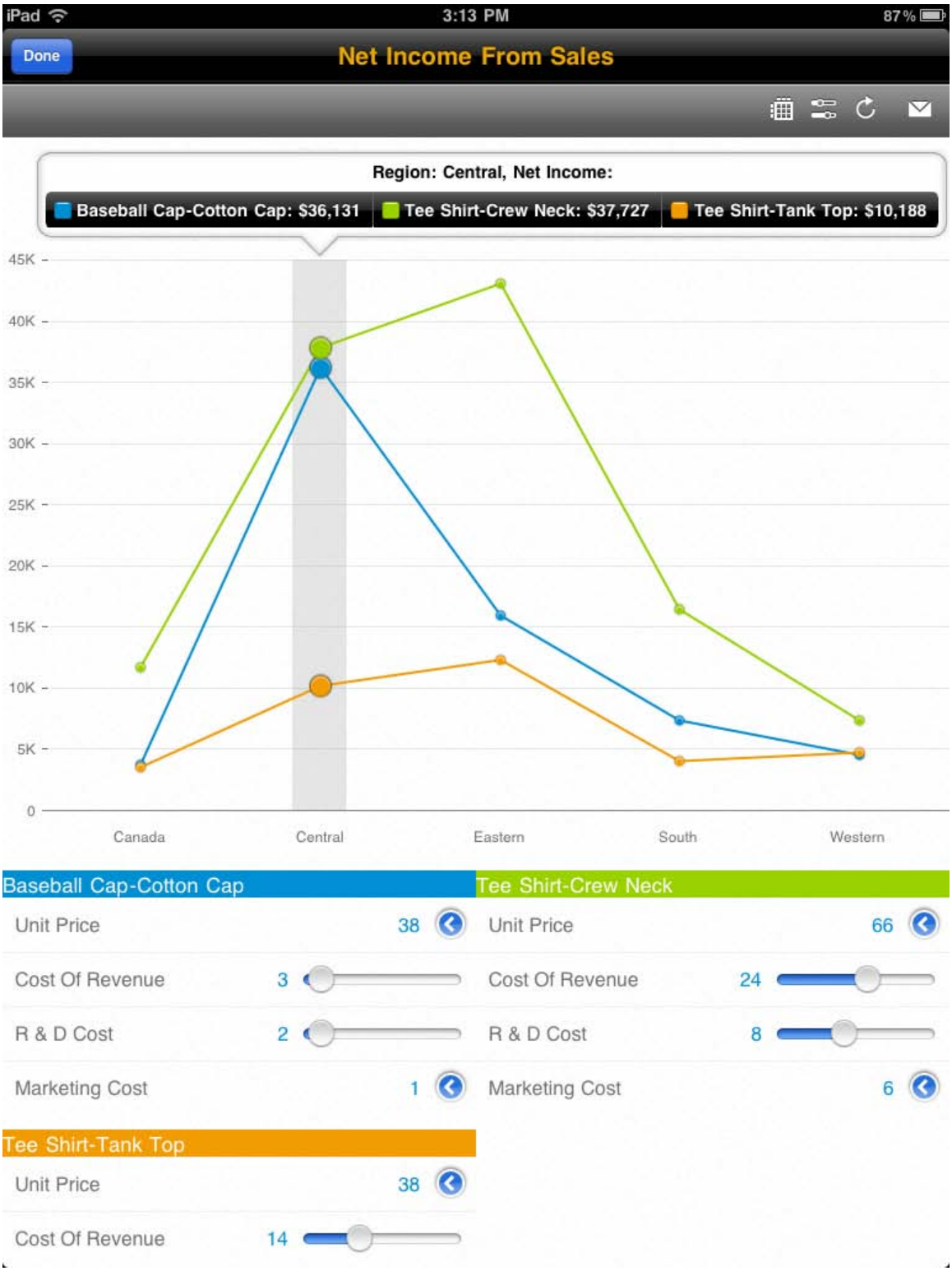
Chart in full-view mode before invoking the What-If panel:



Tap the **What-If** button to invoke the What-If panel for the selected category ("Central" region):



Changing the values in a what-if column updates the chart accordingly:



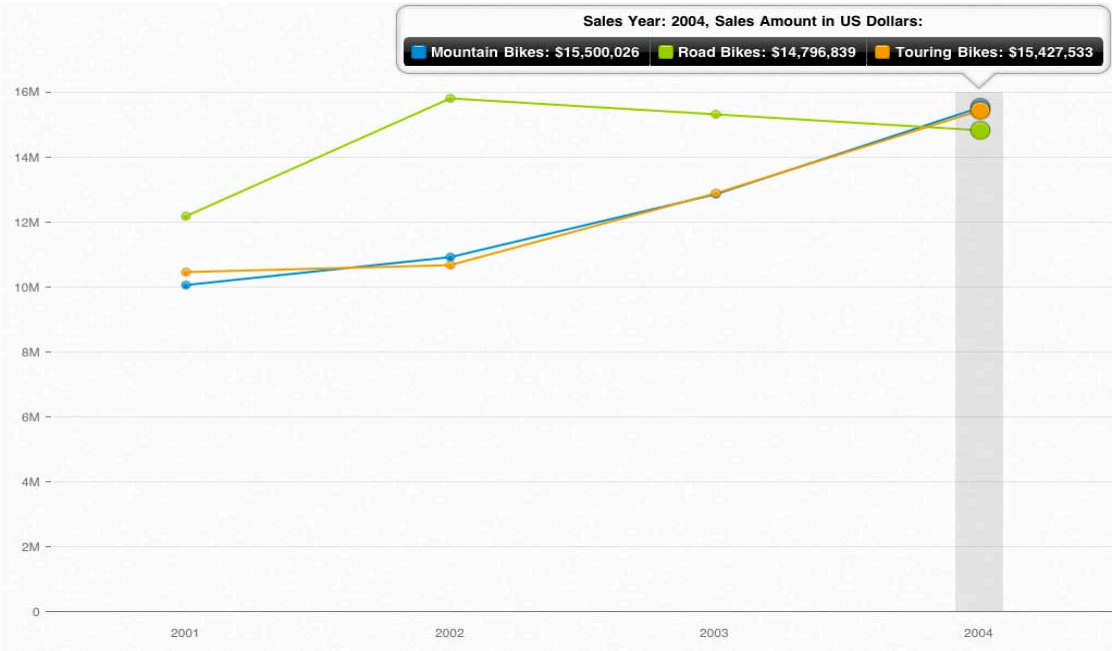
MAAnalyticChartView

Use the MAAnalyticChartView class to create a single-chart view using metadata definitions. Drilldown, semantic zooming, and what-if analysis are all available in the single-chart view.

Example of AnalyticChartView Configuration

The metadata definition is the same as that used to create dashboard view, but only the first chart in the metadata is used. Dashboard layout configurations defined in the metadata are not applicable.

```
<MobileAnalytic >
    <Dashboard>
        <!-- WhatIf Chart Configuration -->
        <Chart Name="WhatIfChart" Title="Net Income From Sales" ChartType="Line" Query="ProductSales">
            <Category Format="" Column="Region" Description="Region" />
            <Series Format="" Column="Product" Description="Product" />
            <Values>
                <Value Format="Currency" Expression="((UnitPrice * Quantity) - ((Cost_Of_Revenue *
Quantity) + (RD_Cost * Quantity) + (Mkt_Cost * Quantity) + (Admin_Cost * Quantity)))"

                ValueSequence="1" Description="Net Income" />
            </Values>
            <!-- What If Columns -->
            <WhatIfColumns>
                <Column Name="UnitPrice" Description="Unit Price" VariableType="float" Value="0.0"
Min="1.0" Max="110.0" Step="1.0" Decimals="2" />
                <Column Name="Cost_Of_Revenue" Description="Cost Of Revenue" VariableType="integer"
Value="0" Min="1" Max="40" />
                <Column Name="RD_Cost" Description="R &amp; D Cost" VariableType="integer" Value="0"
Min="1" Max="18"  />
                <Column Name="Mkt_Cost" Description="Marketing Cost" VariableType="float" Value="0.0"
Min="1.0" Max="10.0" Step="1.0" Decimals="2" />
            </WhatIfColumns>
        </Chart>

    </Dashboard>
</MobileAnalytic>
```



Influence Diagram

An influence diagram is a graphical representation of a decision problem, offering a way to identify and display essential elements, including decisions, uncertainties, and objectives, and how they influence each other.

An influence diagram comprises one default node, and multiple primary nodes and secondary or satellite nodes. Primary nodes directly impact the default node. Each primary node may have multiple associated secondary or satellite nodes. The default mode and all primary nodes are always visible in the model.

Each node in the model shows the measure name, and the current actual value or percentage text.

MAKit influence diagrams include three views: snapshot view, full-screen view in normal mode, and full-screen view in goal-seek mode. It also supports associated charts for some nodes. In full-screen view in goal-seek mode, you can show the cause and effect relationships between nodes as they are defined in the expression attribution of the metadata. You can change the actual value of a node using the keyboard, or by holding and dragging the handle of a node.

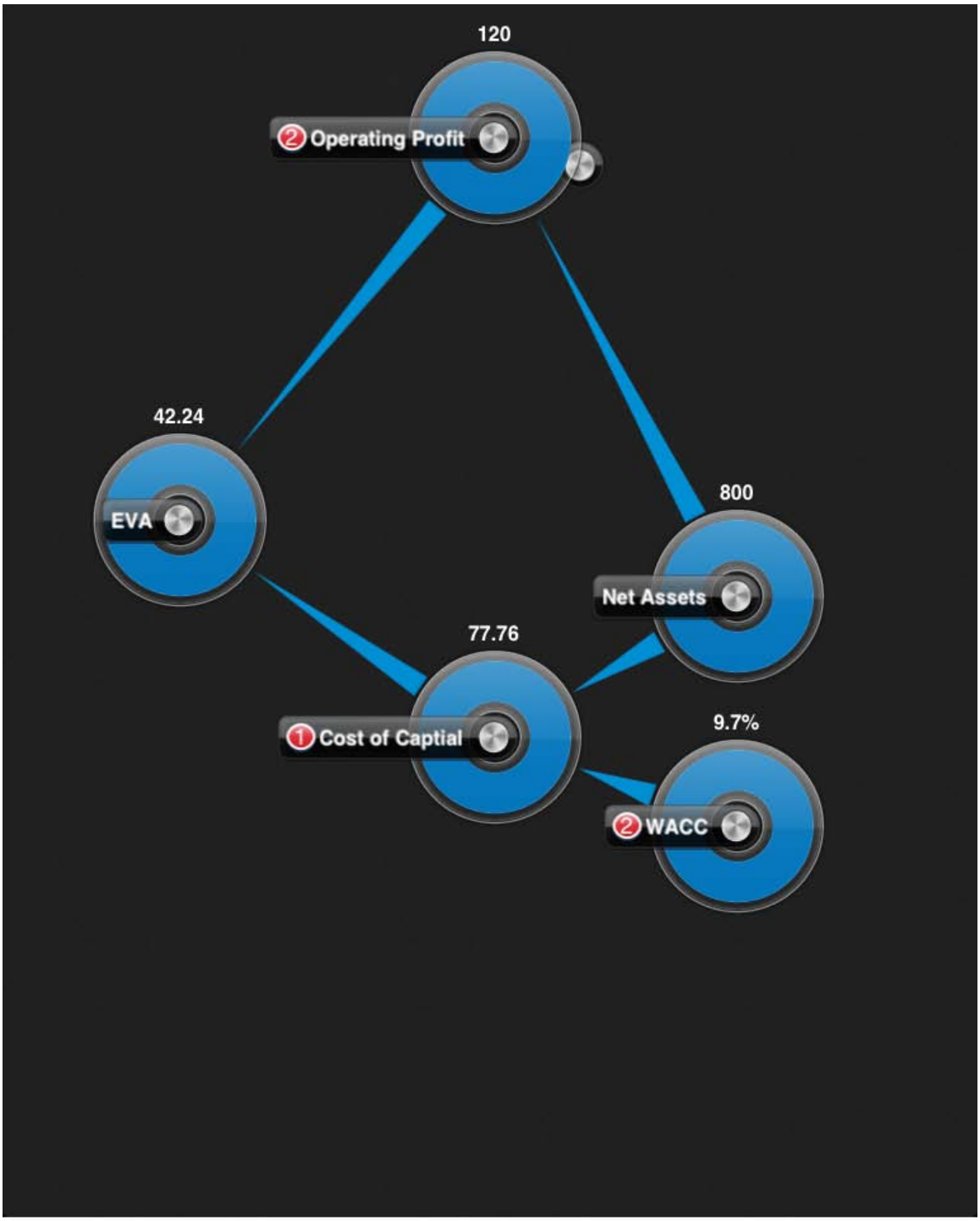
Example of Influence Diagram Configuration

Set the cause and effect relationship between the nodes, and associated charts in metadata.

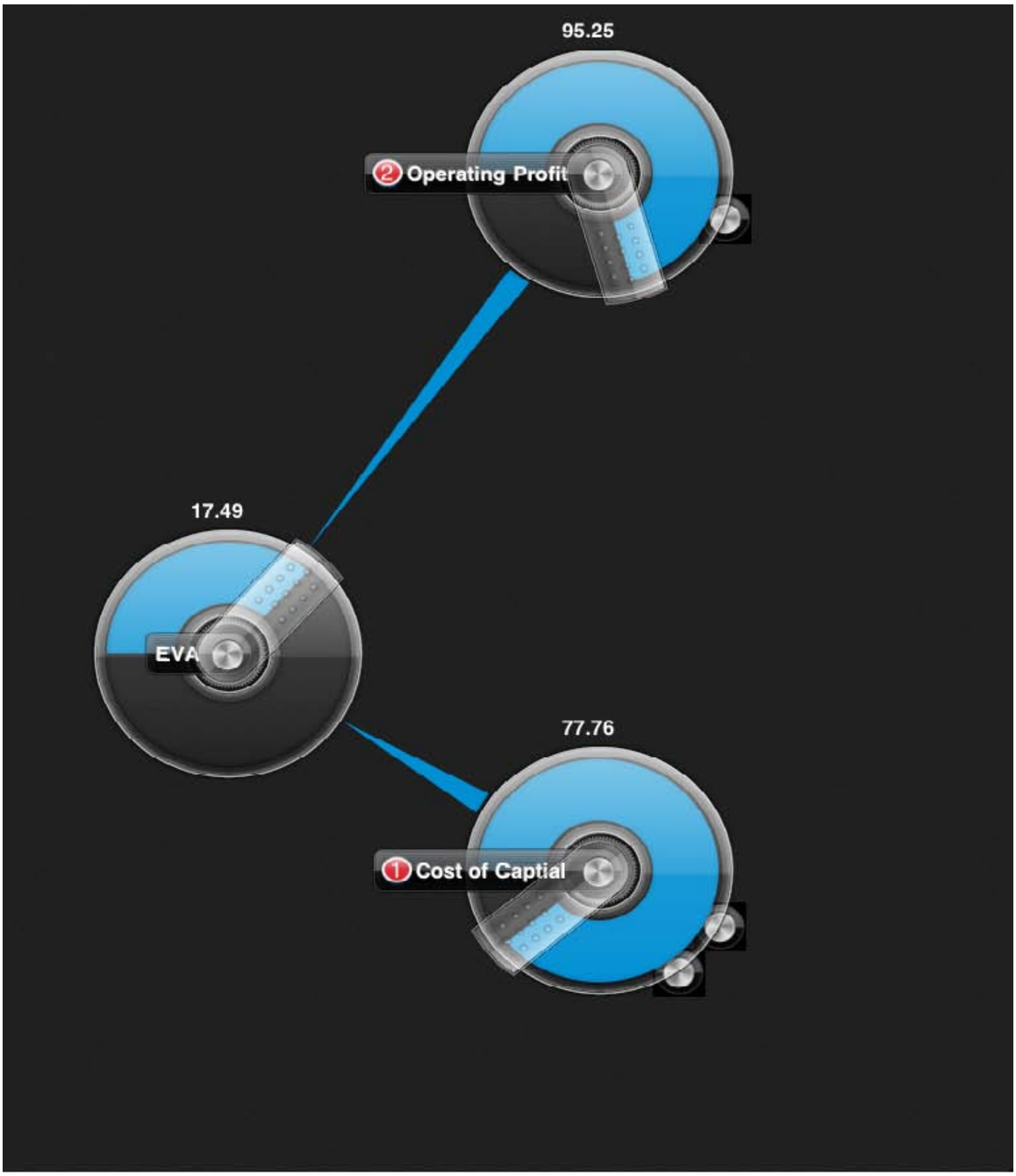
```
<?xml version="1.0" encoding="utf-8"?>
<Diagram Name="EVA-Value-Driver" Query="EVAQuery">
    <Node Name="EVA" Text="EVA" Format="Rounded2" Expression="OperatingProfit1-CostofCaptial"
Default="true"/>
    <Node Name="OperatingProfit1" Text="Operating Profit" ParentNode="EVA"
Expression="RONA*NetAssets1" Format="Rounded0" Charts="chart1,chart2">
        <NodeTo Name="EVA"/>
    </Node>
    <Node Name="CostofCaptial" Text="Cost of Captial" ParentNode="EVA"
Expression="NetAssets1*WACC" Format="Rounded2" Charts="chart1">
        <NodeTo Name="EVA"/>
    </Node>
    <Node Name="RONA" Text="RONA" ParentNode="OperatingProfit1"
Expression="ProfitMargin*NetAssetTurnover" Format="Percentage" Charts="chart2">
        <NodeTo Name="OperatingProfit1"/>
    </Node>
    <Node Name="NetAssets1" Text="Net Assets" ParentNode="OperatingProfit1,CostofCaptial"
Format="Rounded0">
        <NodeTo Name="OperatingProfit1"/>
        <NodeTo Name="CostofCaptial"/>
    </Node>
    <Node Name="WACC" Text="WACC" ParentNode="CostofCaptial" Format="Percentage">
        <NodeTo Name="CostofCaptial"/>
    </Node>
   <Node Name="ProfitMargin" Text="Profit Margin" ParentNode="RONA"
Expression="OperatingProfit2/Sales1" Format="Percentage">
        <NodeTo Name="RONA"/>
    </Node>
    <Node Name="NetAssetTurnover" Text="Total Net Asset Turnover" ParentNode="RONA"
Expression="Sales1/NetAssets2" Format="Rounded2">
        <NodeTo Name="RONA"/>
    </Node>
    <Node Name="OperatingProfit2" Text="Operating Profit" ParentNode="ProfitMargin"
Expression="EBIT*IncomeTax" Format="Rounded0">
        <NodeTo Name="ProfitMargin"/>
    </Node>
    <Node Name="Sales1" Text="Sales" ParentNode="ProfitMargin,NetAssetTurnover"
Format="Rounded0">
        <NodeTo Name="ProfitMargin"/>
        <NodeTo Name="NetAssetTurnover"/>
```

```
    </Node>
    <Node Name="NetAssets2" Text="Net Assets" ParentNode="NetAssetTurnover"
Expression="Assets-Liabilities" Format="Rounded0">
        <NodeTo Name="NetAssetTurnover"/>
    </Node>
    <Node Name="EBIT" Text="EBIT" ParentNode="OperatingProfit2" Expression="Contribution-
Overhead" Format="Rounded0">
        <NodeTo Name="OperatingProfit2"/>
    </Node>
    <Node Name="IncomeTax" Text="(1-Rate of Income Tax)" ParentNode="OperatingProfit2"
Format="Rounded2">
        <NodeTo Name="OperatingProfit2"/>
    </Node>
    <Node Name="Assets" Text="Assets" ParentNode="NetAssets2"
Expression="FixedAssets+CurrentAssets" Format="Rounded0">
        <NodeTo Name="NetAssets2"/>
    </Node>
    <Node Name="Liabilities" Text="Interest Free Liabilities" ParentNode="NetAssets2"
Format="Rounded0">
        <NodeTo Name="NetAssets2"/>
    </Node>
    <Node Name="Contribution" Text="Contribution" ParentNode="EBIT" Expression="Sales2-
GoodsSold" Format="Rounded0">
        <NodeTo Name="EBIT"/>
    </Node>
    <Node Name="Overhead" Text="Overhead" ParentNode="EBIT" Format="Rounded0">
        <NodeTo Name="EBIT"/>
    </Node>
    <Node Name="FixedAssets" Text="Fixed Assets" ParentNode="Assets" Format="Rounded0">
        <NodeTo Name="Assets"/>
    </Node>
    <Node Name="CurrentAssets" Text="Current Assets" ParentNode="Assets" Format="Rounded0">
        <NodeTo Name="Assets"/>
    </Node>
    <Node Name="Sales2" Text="Sales" ParentNode="Contribution" Format="Rounded0">
        <NodeTo Name="Contribution"/>
    </Node>
    <Node Name="GoodsSold" Text="Variable Cost of Goods Sold" ParentNode="Contribution"
Format="Rounded0">
        <NodeTo Name="Contribution"/>
    </Node>
 </Diagram>
```

Snapshot view:



Full-screen view in goal-seek mode:



Predictive Analysis and Threshold Reference

Predictive analysis shows forecasted or projected data points using a dashed-line pattern. Threshold values highlight values that fall outside of a specified range; for example, you can assign a reference value, which appears on the y-axis, to highlight some benchmark value.

Predictive analysis and threshold references are supported only in line charts. Configure the required parameters for predictive analysis and threshold reference with the corresponding values.

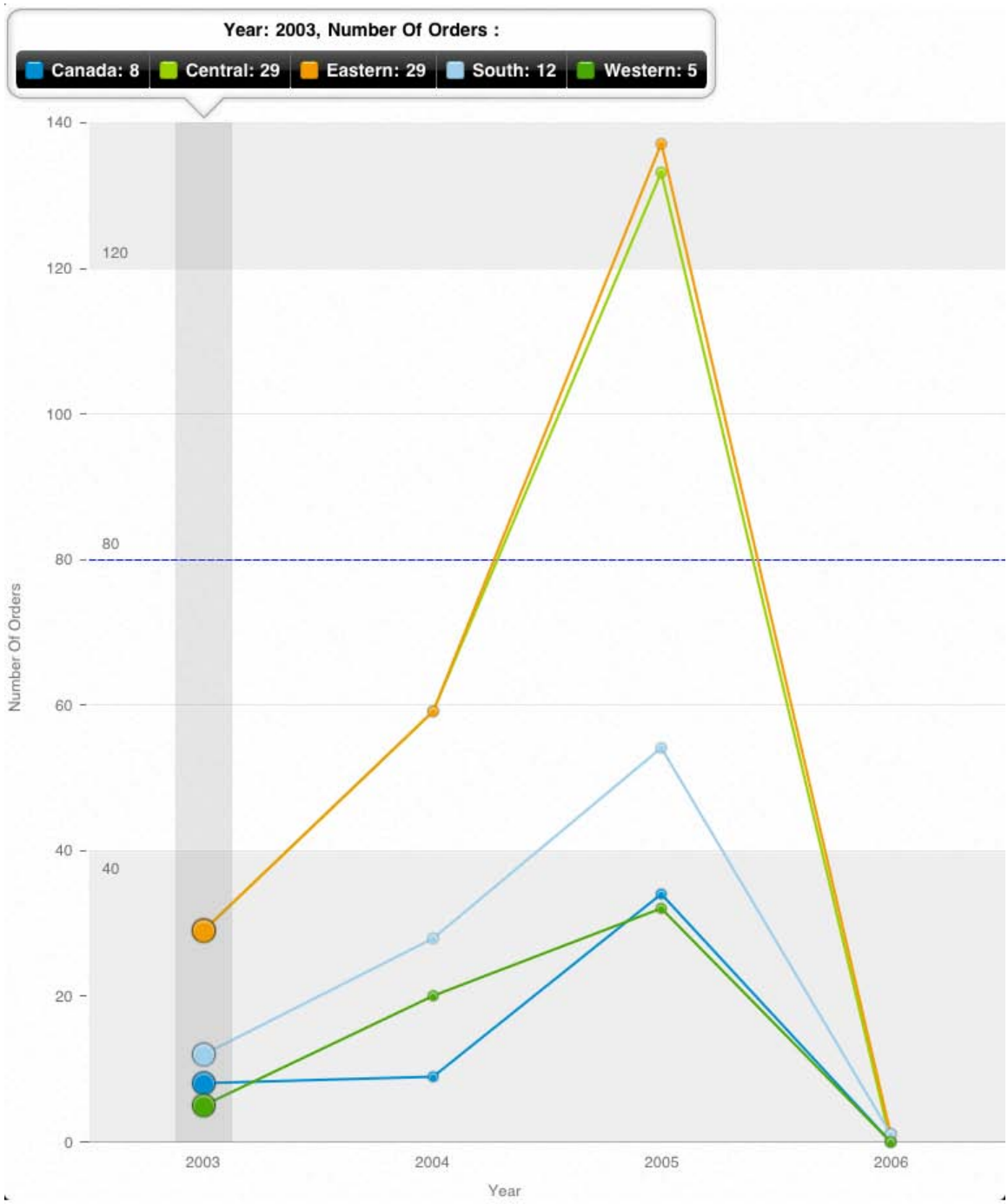
Configure these variables for threshold reference and predictive analysis:

- referenceValue
- upperThreshold
- lowerThreshold
- predictiveStartCategory

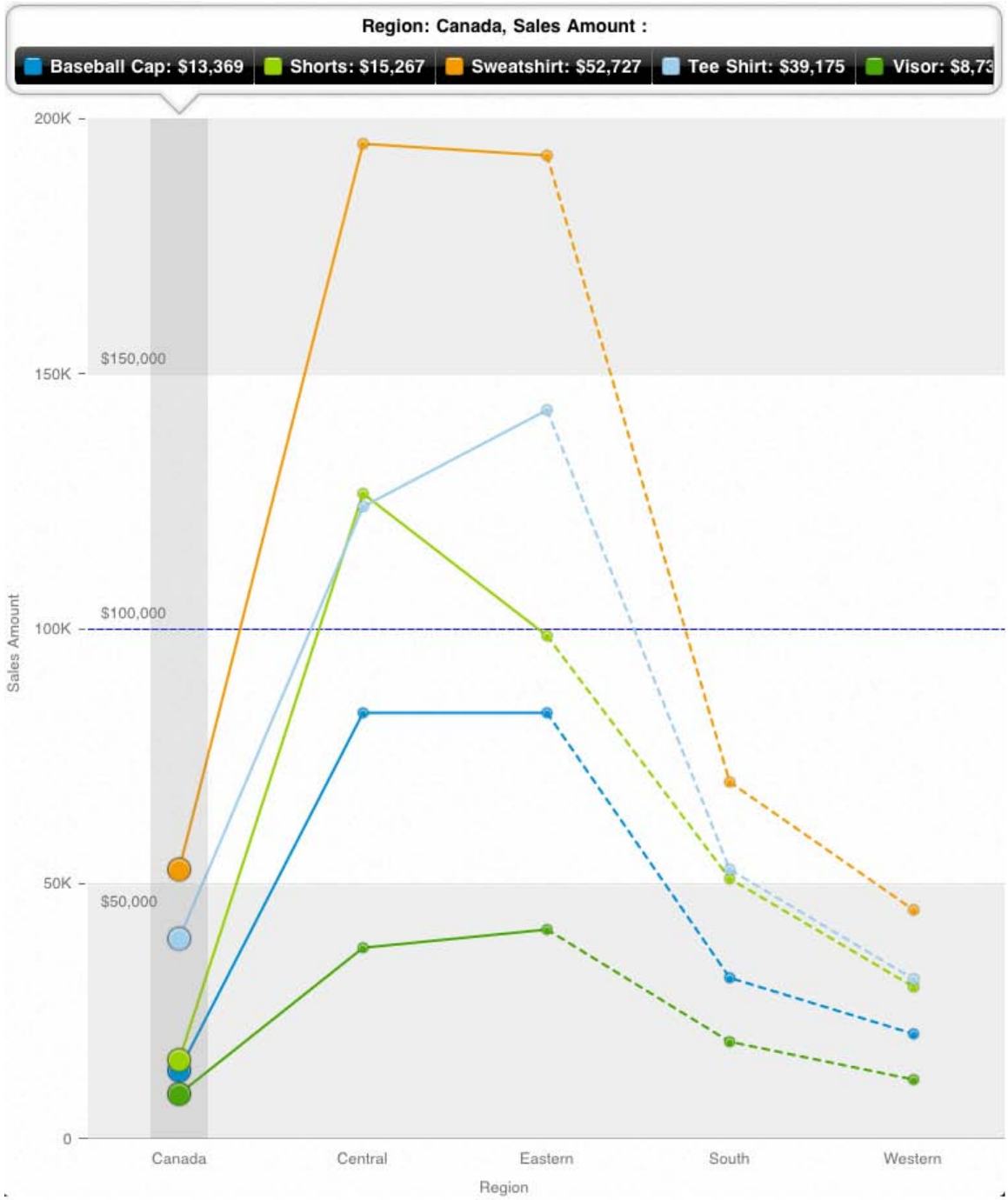
Example of Threshold and Predictive Analysis Configuration

```
// This is for Predictive Analysis and Threshold Reference

    else if ([queryName isEqualToString:@"SalesOrder_Pred"]) {
    SqliteDataSource *s1 = [[SqliteDataSource alloc] init];
    table.chartData = [s1 SalesOrderSummary];

    table.referenceValue = [[[NSNumber alloc]initWithDouble:100000]autorelease];
    table.upperThreshold = [[[NSNumber alloc]initWithDouble:150000]autorelease];
    table.lowerThreshold = [[[NSNumber alloc]initWithDouble:50000]autorelease];

    table.predictiveStartCategory = @"Eastern";


    [s1 release];
    }


   // This is for Predictive Analysis
    else if ([queryName isEqualToString:@"NumberOfOrders_Pred"]) {
        SqliteDataSource *s1 = [[SqliteDataSource alloc] init];
        table.chartData = [s1 NumberOfOrdersByYear];
        [s1 release];


        table.referenceValue = [[[NSNumber alloc]initWithDouble:80]autorelease];
        table.upperThreshold = [[[NSNumber alloc]initWithDouble:120]autorelease];
        table.lowerThreshold = [[[NSNumber alloc]initWithDouble:40]autorelease];
```

This example shows a line chart with a threshold reference. Threshold references highlight the benchmark values and the values that comprise the upper and lower thresholds. Benchmark values use a dotted blue horizontal line, whereas the upper and lower threshold values are shaded with reference to the y-axis.



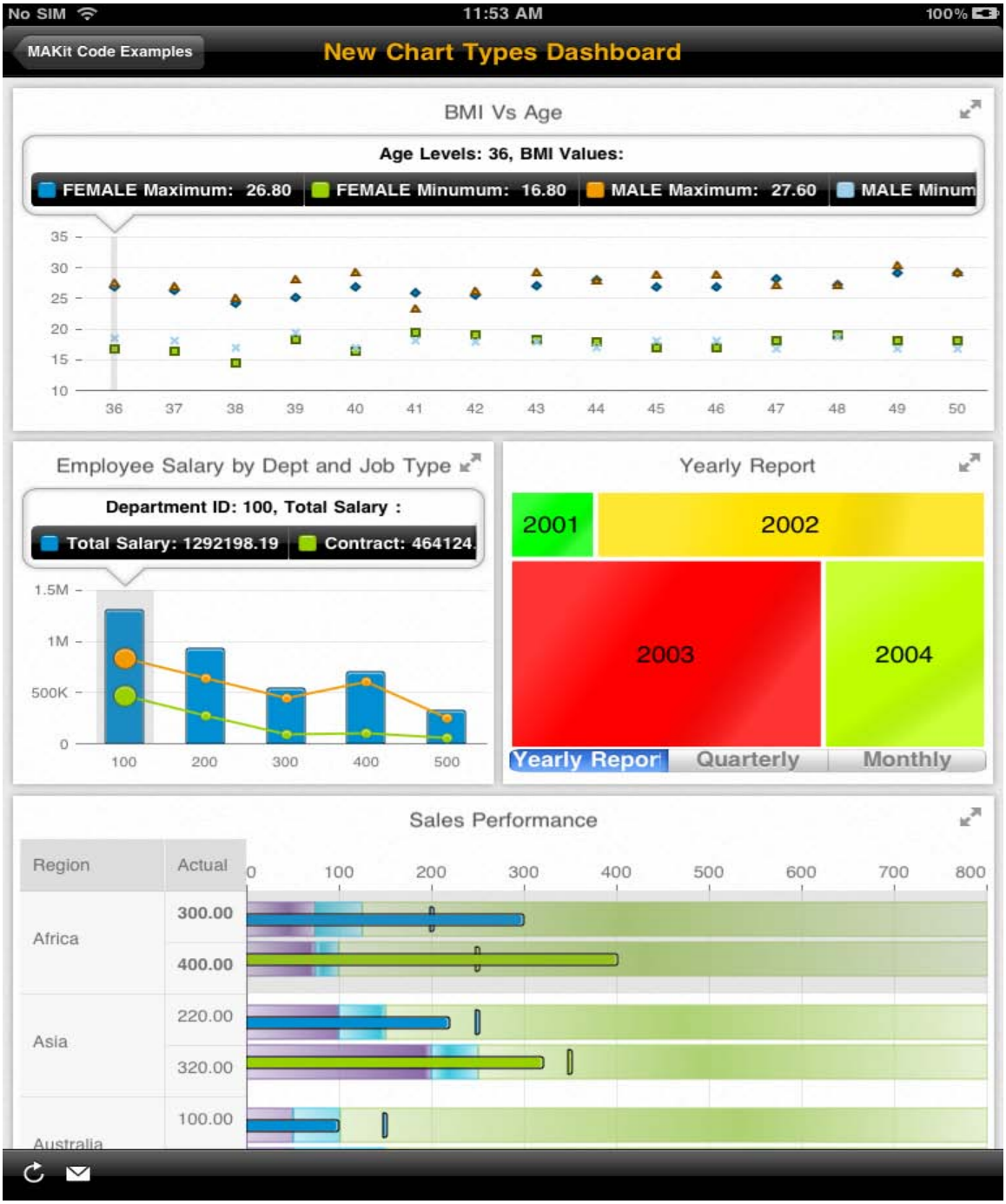
In this example, a line chart that shows predictive analysis and a threshold reference, the predictive line begins at the “Eastern” category, which is also shown in the example metadata:



MAKit 2.2 New Features

This section demonstrates the new chart types supported by MAKit 2.2. The new chart types are grouped together to create a sample dashboard. See the `Dashboard_NewChartTypes.xml` metadata file to learn how to configure different chart types.

This dashboard layout configuration renders a chart similar to:



Examples of New Chart Configuration

```
<Chart Name="ScatterChart" Title="BMI Vs Age" ChartType="Scatter" Query="bmi">
   <Category Format="Rounded0" Column="age" DisplayName="Age Levels" />
   <Series Format="" Column="sex" DisplayName="Sex" />
   <Values>
       <Value Format="Rounded2" Expression="bmi" ValueSequence="1"
DisplayName="BMI Values" />
   </Values>
</Chart>

<Chart Name ="BulletChart" Title="Sales Performance" ChartType="HorizontalBullet"
Query="performance">
   <Category Format="" Column="region" DisplayName="Region"/>
   <Series Format="" Column="sex" DisplayName="Male/Female" />
   <Values>
        <Value Expression = "actual"  ValueSequence = "1" DisplayName="Actual"/>
        <Value Expression = "bad" ValueSequence = "2" DisplayName="Bad"/>
        <Value Expression = "satisfactory" ValueSequence = "3"
DisplayName="Satisfactory"/>
        <Value Expression = "target" ValueSequence = "4" DisplayName="Target"/>
   </Values>
</Chart>

<TreeMapGroup Name="TreemapChart" Title="Detailed Sales Report">
   <TreeMapLevel Name="By Year" Title="Yearly Report" ChartType="TreeMap"
ChartSequence="1" Query="ResellerForYear">
   <Category Format="" Column="SalesYear" DisplayName="Company" />
   <Values>
        <Value Format="Rounded2" Expression="SalesAmount" ValueSequence="1"
DisplayName="Sales Amount" />
        <Value Format="Rounded2" Expression="MfgCost" ValueSequence="2"
DisplayName="Manufacturing Cost" />
   </Values>
   </TreeMapLevel>

   <TreeMapLevel Name="By Quarter" Title="Quarter" ChartType="TreeMap"
ChartSequence="2" Query="ResellerForQtr">
   <Category Format="" Column="Quarter" DisplayName="Quarter" />
   <Values>
        <Value Format="Rounded2" Expression="SalesAmount" ValueSequence="1"
DisplayName="Sales Amount" />
        <Value Format="Rounded2" Expression="MfgCost" ValueSequence="2"
DisplayName="Manufacturing Cost" />
   </Values>
   </TreeMapLevel>

   <TreeMapLevel Name="By Month" Title="Month" ChartType="TreeMap"
ChartSequence="3" Query="ResellerForMonth">
   <Category Format="" Column="Month" DisplayName="Month" />
   <Values>
        <Value Format="Rounded2" Expression="SalesAmount" ValueSequence="1"
DisplayName="Sales Amount" />
        <Value Format="Rounded2" Expression="MfgCost" ValueSequence="2"
DisplayName="Manufacturing Cost" />
    </Values>
    </TreeMapLevel>
</TreeMapGroup>

<OverlayGroup Name="OverlayChart" Title="Employee Salary Comparision">
    <Category Format="Rounded0" Column="dept_id" DisplayName="Department ID" />
    <Layer Name="Employee Salary" Title="Employee Salary by Dept and Job Type"
ChartType="Column" ChartSequence="1" Query="EmpSalaryRpt">

    <Values>
        <Value Format="Rounded2" Expression="salary" ValueSequence="1"
DisplayName="Total Salary" />
        </Values>
    </Layer>
```

```
    <Layer Name="ResellerByYear" Title="Revenue by Year" ChartType="Line"
ChartSequence="2" Query="EmpSalary_Jobtype">
    <Series Format="" Column="status" DisplayName="Job Type" />
    <Values>
        <Value Format="Rounded2" Expression="salary" ValueSequence="1"
DisplayName="Salary(Job Type)" />
     </Values>
     </Layer>
</OverlayGroup>
```

The metadata file also configures the dashboard layouts for either portrait or landscape mode.

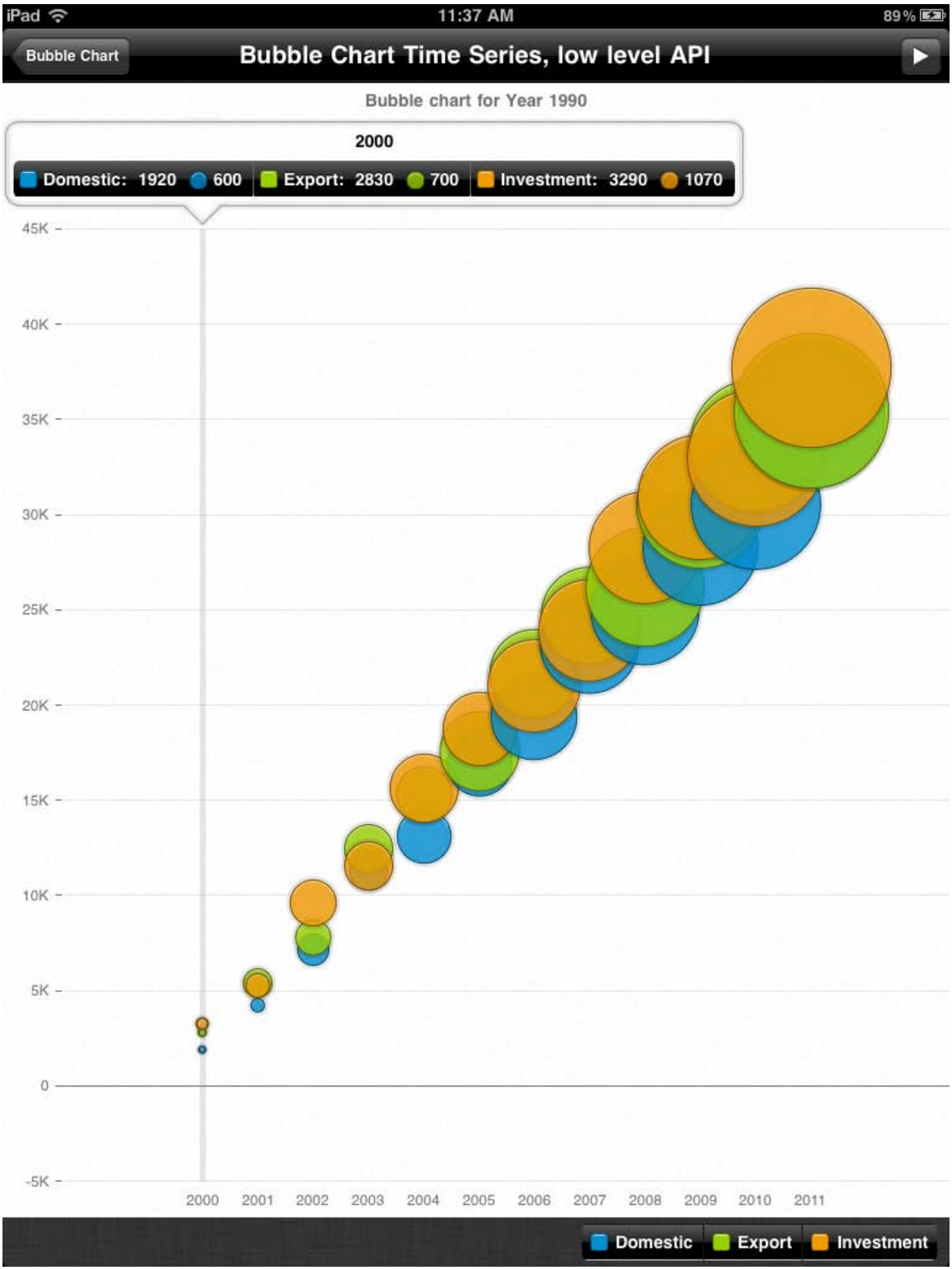
Example of Dashboard Layout

```
<DashboardLayout>
      <Landscape>
          <Layout Name="ScatterChart" Top="1" Left="1" Width="98" Height="32" />
          <Layout Name="OverlayChart" Top="34" Left="1" Width="48" Height="32" />
          <Layout Name="TreemapChart" Top="34" Left="50" Width="49" Height="32" />
          <Layout Name="BulletChart" Top="67" Left="1" Width="98" Height="35" />
      </Landscape>

      <Portrait>
          <Layout Name="ScatterChart" Top="1" Left="1" Width="98" Height="32" />
          <Layout Name="OverlayChart" Top="34" Left="1" Width="48" Height="32" />
          <Layout Name="TreemapChart" Top="34" Left="50" Width="49" Height="32" />
          <Layout Name="BulletChart" Top="67" Left="1" Width="98" Height="35" />
      </Portrait>
</DashboardLayout>
```

Time-Series Drawing

Use the **reloadData** method to create an animated time series chart. When **reloadData** is called, the chart queries its datasource for a fresh set of data. This code snippet illustrates how **reloadData** has been invoked within a timer thread to simulate a time series from 1990 to 2012:

```
- (IBAction)play:(id)sender
{
    if (self.timer)
    {
        [self.timer invalidate];
        self.timer = nil;
        self.year = 1990;
    }

    self.chart.showsSelectedDataLabel = NO;
    self.timer = [NSTimer scheduledTimerWithTimeInterval:0.3
                                    target:self
                                 selector:@selector(reloadDataInBackground)
                                 userInfo:nil
                                  repeats:YES];
}
```

```
- (void)reloadDataInBackground
{
   NSAutoreleasePool* pool = [[NSAutoreleasePool alloc] init];
       @synchronized(self)
       {
        self.year++;
        [self.chart reloadData:NO];

        if (self.year >= 2012)
        {
            [self.timer invalidate];
            self.chart.showsSelectedDataLabel = YES;
        }

    }
    [pool release];
}
```

This chart shows a time series drawing of bubble chart: