WickedCharts Wicket Component
This component (isis-wicket-wickedcharts) integrates Wicked Charts. Wicked Charts is in turn an integration between Apache Wicket and the Highcharts JS charting library).

Please note that while this project and Wicked Charts are licensed under Apache 2.0 License, Highcharts itself is only free for non-commercial use. See the Highcharts website for further details.

There are in fact two separate components:

• **summarycharts**: render a standalone collection with BigDecimal properties as a chart. (This component can be thought of as an enhancement of the base summary view provided by Isis Wicket viewer).

• **scalarchart**: renders a standalone scalar value (from an action invocation) as a chart
Screenshots

The module's functionality can be explored by running the quickstart with example usage using the org.incode.domainapp.example.app.modules.ExampleDomWktSummerNoteAppManifest.

A home page is displayed when the app is run:

Note that the example entity (todo item) has two numeric (BigDecimal) properties:
Summary Charts

Invoking an action that returns a collection of entities:

... shows an additional button to view those entities in a summary chart:

Clicking on the button renders a chart where the values of all numeric (BigDecimal) properties are plotted:
Scalar Charts

Arbitrary charts can be returned from any action. For example this action:

... renders a pie chart splitting out the example Todo entities by their category:
API & Usage

Summary Charts

There is no special usage; a standalone collection of any entity with one or more properties of type `BigDecimal` will be rendered using the `summarycharts` extension.

Scalar Chart

Any action returning the `WickedChart` value type should be rendered as a chart. The `WickedChart` value type is simply a wrapper around the wicked chart's `Options` class:

```java
import com.googlecode.wickedcharts.highcharts.options.Options;

public class WickedChart implements Serializable {
    private Options options;

    public WickedChart(Options options) { ... }

    ...  
}
```

Any chart supported by Wicked Charts (see their showcase app) should work.
How to configure/use

Classpath

Add the component to your project's dom module's pom.xml:

```xml
<dependency>
  <groupId>org.isisaddons.wicket.wickedcharts</groupId>
  <artifactId>isis-wicket-wickedcharts-cpt</artifactId>
</dependency>
```

Check for later releases by searching Maven Central Repo.
Known issues / Limitations

Although the WickedChart class (in the 'scalarchart's API) has value semantics, it will (currently) not render as a chart if used as an entity property. Such a property should be persistable, however. Therefore a workaround is to hide the property and instead provide an action to show the chart.

For example:

```java
public class MyEntity {

    private WickedChart chart;
    @Property(hidden=EVERYWHERE)
    public WickedChart getChart() { ... }
    public void setChart(WickedChart chart) { ... }

    public WickedChart showChart() {
        return getChart();
    }
    ...
}
```
Dependencies

Maven can report modules dependencies using:

```
mvn dependency:list -o -pl modules/wkt/wickedcharts/impl -D excludeTransitive=true
```

which, excluding Apache Isis itself, returns these compile/runtime dependencies:

```
commons-codec:commons-codec:jar:1.7
com.googlecode.wicked-charts:highcharts-wrapper:jar:2.0.1
com.googlecode.wicked-charts:wicked-charts-wicket7:jar:2.0.1
```

For further details on 3rd-party dependencies, see:

- [Apache Commons (Codec)](#)
- [Wicked Charts](#)

It also includes a Javascript dependency on [Highcharts](#).

⚠️ Note that the [Highchart License](#) is commercial and must be purchased (unless in use for a personal/open source project).