

intra-mart Accel Platform

グラフ描画モジュールプログラミングガイド

2012/10/01 初版

<< 変更履歴 >>

変更年月日	変更内容
2012/10/01	初版

<< 目次 >>

1	はじめに	1
2	API	2
2.1	概要	2
2.2	JavaEE開発モデル	2
2.2.1	タグライブラリ	2
2.2.2	im-BizAPI	2
2.3	スクリプト開発モデル	2
2.3.1	タグライブラリ	2
3	JavaEE開発モデル	4
3.1	はじめに	4
3.2	円グラフ	4
3.2.1	サンプルで利用するAPI	4
3.2.2	サンプルコード	4
3.2.3	実行結果	5
3.3	ポートフォリオ	6
3.3.1	サンプルで利用するAPI	6
3.3.2	サンプルコード	6
3.3.3	実行結果	7
3.4	複合グラフ その1	8
3.4.1	サンプルで利用するAPI	8
3.4.2	サンプルコード	8
3.4.3	実行結果	9
3.5	複合グラフ その2	10
3.5.1	サンプルで利用するAPI	10
3.5.2	サンプルコード	10
3.5.3	実行結果	11
4	スクリプト開発モデル	12
4.1	はじめに	12
4.2	円グラフ	12
4.2.1	サンプルで利用するAPI	12
4.2.2	サンプルコード	12
4.2.3	実行結果	13
4.3	ポートフォリオ	14
4.3.1	サンプルで利用するAPI	14
4.3.2	サンプルコード	14
4.3.3	実行結果	15
4.4	複合グラフ その1	16
4.4.1	サンプルで利用するAPI	16
4.4.2	サンプルコード	16
4.4.3	実行結果	17
4.5	複合グラフ その2	18
4.5.1	サンプルで利用するAPI	18
4.5.2	サンプルコード	18
4.5.3	実行結果	19
5	設定ファイルと設定	20
5.1	WEB-INF/conf/graph/default.properties	20

5.1.1	ディレクトリ構造.....	20
5.1.2	設定項目	20
5.2	WEB-INF/web.xml.....	23
5.2.1	jp.co.intra_mart.foundation.chart.servlet.GraphDrawerServletクラス	23
5.3	テーマ(Theme)機能	23
5.3.1	利用手順	23
6	ディレクトリ構造.....	25
6.1	ディレクトリ構成.....	25
7	付録	26
7.1	グラフサンプル	26
7.1.1	メニュー	26
7.1.2	ディレクトリ構成.....	26

1 はじめに

intra-mart Accel Platform（以下 iAP）のグラフ描画モジュールでは、グラフ作成ライブラリである

「JfreeChart (<http://www.jfree.org/jfreechart/>)」を利用してグラフの描画を実現します。

本ドキュメントは、グラフ描画モジュールの基本的な設定及びプログラミングについて解説します。

2 API

2.1 概要

iAP では、以下グラフ描画モジュール API が提供されています。

2.2 JavaEE開発モデル

2.2.1 タグライブラリ

intra-mart 標準 タグライブラリ	
グラフ描画タグライブラリ	imchart:chart

2.2.2 im-BizAPI

アプリケーション共通モジュール API	
パッケージ	jp.co.intra_mart.foundation.chart.data
	jp.co.intra_mart.foundation.chart.dataset
	jp.co.intra_mart.foundation.chart.drawer
	jp.co.intra_mart.foundation.chart.graphInfo

※ 上記パッケージに含まれるクラス、メソッドの詳細については API リストを参照してください。

2.3 スクリプト開発モデル

2.3.1 タグライブラリ

画面共通 IMART タグライブラリ	
グラフ描画系タグ	chart

3 JavaEE 開発モデル

3.1 はじめに

ここでは、JavaEE 開発モデルでのグラフ描画モジュールの利用方法(プログラミング)を、簡単なサンプルを例に解説していきます。

3.2 円グラフ

3.2.1 サンプルで利用するAPI

グラフ描画タグライブラリ	imchart:chart
style 属性	pie

アプリケーション共通モジュール API	jp.co.intra_mart.foundation.chart.graphInfo.PieGraphInfo
	jp.co.intra_mart.foundation.chart.dataset.PieDataset
	jp.co.intra_mart.foundation.chart.data.PieData

3.2.2 サンプルコード

```
<jsp>
<%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>

<%@ taglib uri="http://www.intra-mart.co.jp/taglib/foundation/chart" prefix="imchart"%>
<%@page import="jp.co.intra_mart.foundation.chart.dataset.*"%>
<%@page import="jp.co.intra_mart.foundation.chart.data.*"%>
<%@page import="jp.co.intra_mart.foundation.chart.graphInfo.*"%>

<%
// 円グラフ用サンプルデータ
PieGraphInfo pieGraphInfo = new PieGraphInfo();
PieDataset pieDataset = new PieDataset();
pieDataset.addDataset(new PieData("東京支店", 700));
pieDataset.addDataset(new PieData("大阪支店", 300));
pieDataset.addDataset(new PieData("神奈川支店", 250));
pieDataset.addDataset(new PieData("千葉支店", 302));
pieDataset.addDataset(new PieData("名古屋支店", 155));
pieDataset.addDataset(new PieData("埼玉支店", 400));
pieGraphInfo.setDataset(pieDataset);
%>
<div class="imui-title">
<h1>円グラフ サンプル</h1>
</div>
<div class="imui-form-container-narrow">
<table>
<tr>
<td><imchart:chart displayType="3D" style="pie" data="<%=pieGraphInfo %>"></imchart:chart></td>
</tr>
</table>
</div>
```

図解説：コードの構造を示す。左側は「円グラフ用データ生成部分」、右側は「グラフ画像表示部分」である。

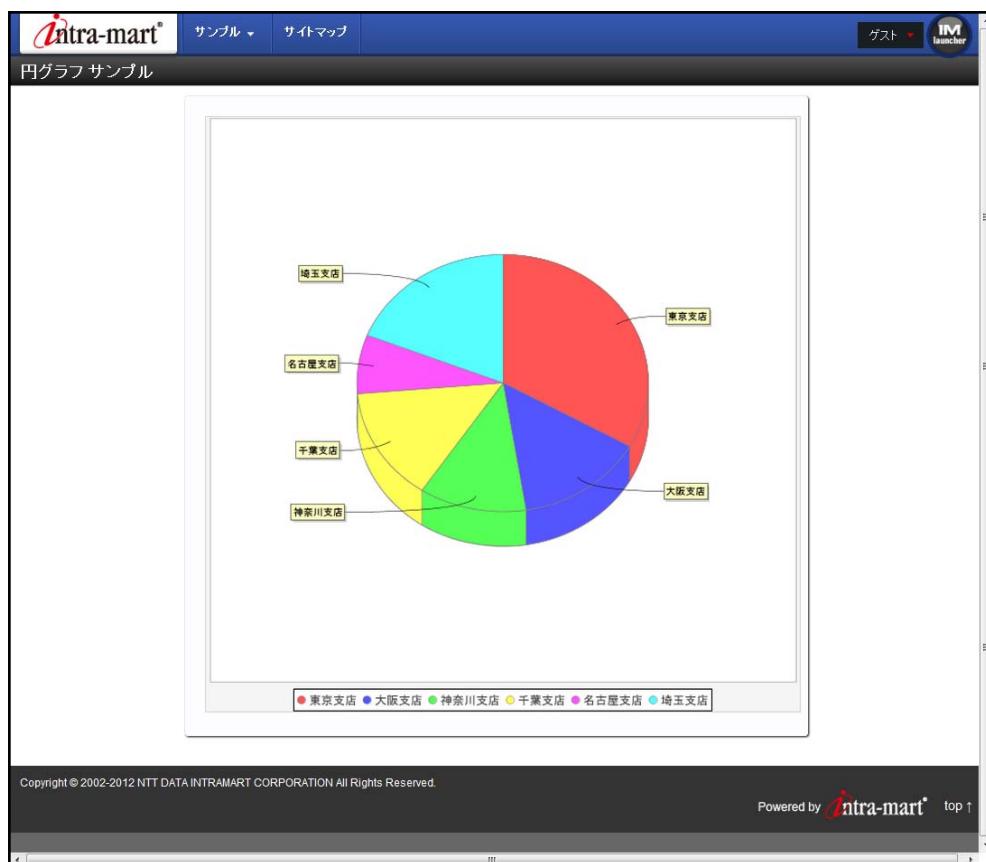
円グラフ用データ生成部分（左側）：

- PieGraphInfo pieGraphInfo = new PieGraphInfo();
- PieDataset pieDataset = new PieDataset();
- pieDataset.addDataset(new PieData("東京支店", 700));
- pieDataset.addDataset(new PieData("大阪支店", 300));
- pieDataset.addDataset(new PieData("神奈川支店", 250));
- pieDataset.addDataset(new PieData("千葉支店", 302));
- pieDataset.addDataset(new PieData("名古屋支店", 155));
- pieDataset.addDataset(new PieData("埼玉支店", 400));
- pieGraphInfo.setDataset(pieDataset);

グラフ画像表示部分（右側）：

- <imchart:chart displayType="3D" style="pie" data="<%=pieGraphInfo %>"></imchart:chart>

3.2.3 実行結果



3.3 ポートフォリオ

3.3.1 サンプルで利用するAPI

グラフ描画タグライブラリ	imchart:chart
style 属性	portFolio

アプリケーション共通モジュール API	jp.co.intra_mart.foundation.chart.graphInfo.PortFolioGraphInfo
	jp.co.intra_mart.foundation.chart.dataset.PortFolioDataset
	jp.co.intra_mart.foundation.chart.data.PortFolioData
	jp.co.intra_mart.foundation.chart.data.PortFolioValue

3.3.2 サンプルコード

```
<jsp>
<%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>

<%@ taglib uri="http://www.intra-mart.co.jp/taglib/foundation/chart" prefix="imchart"%>
<%@page import="jp.co.intra_mart.foundation.chart.dataset.*"%>
<%@page import="jp.co.intra_mart.foundation.chart.data.*"%>
<%@page import="jp.co.intra_mart.foundation.chart.graphInfo.*"%>

<%
// ポートフォリオグラフ用サンプルデータ
PortFolioGraphInfo portFolioInfo = new PortFolioGraphInfo(50, 6000);
PortFolioDataset portFolioDataset = new PortFolioDataset();

PortFolioData pfSeries1 = new PortFolioData();
pfSeries1.setName("東京");
pfSeries1.addValue(new PortFolioValue(15, 3000, 1000));
pfSeries1.addValue(new PortFolioValue(90, 9000, 1000));

PortFolioData pfSeries2 = new PortFolioData();
pfSeries2.setName("大阪");
pfSeries2.addValue(new PortFolioValue(50, 5000, 3000));

PortFolioData pfSeries3 = new PortFolioData();
pfSeries3.setName("名古屋");
pfSeries3.addValue(new PortFolioValue(35, 3000, 2000));

portFolioDataset.addData(pfSeries1);
portFolioDataset.addData(pfSeries2);
portFolioDataset.addData(pfSeries3);

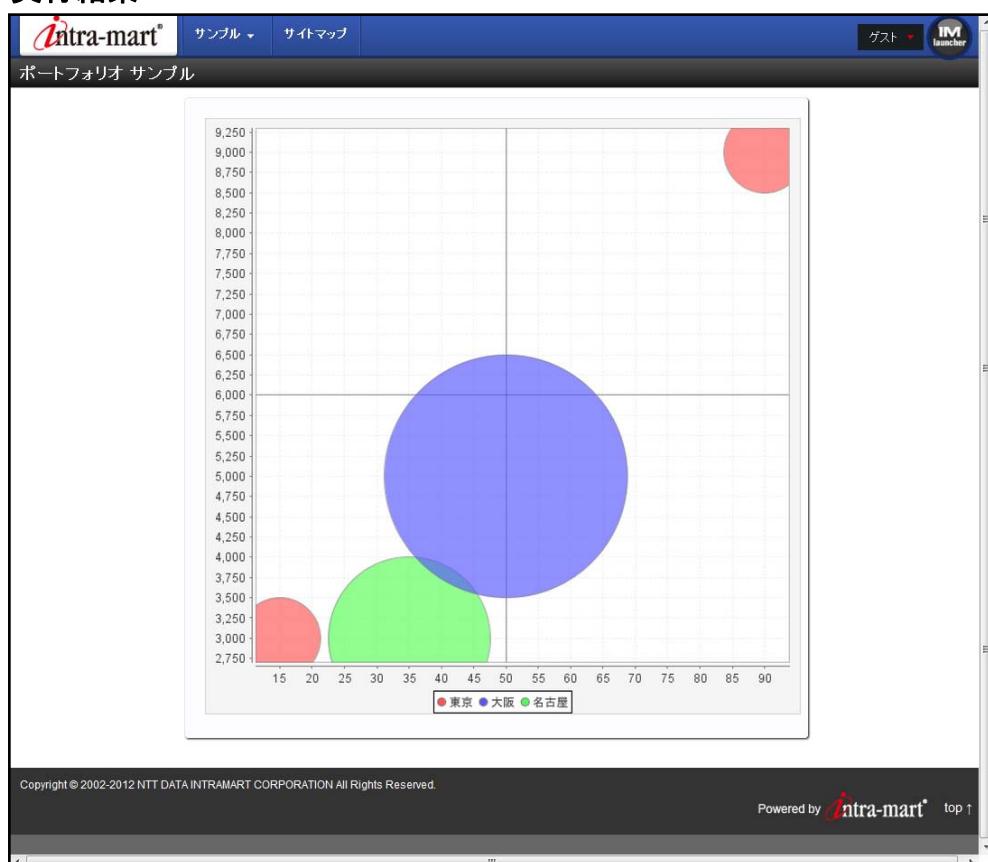
portFolioInfo.setDataset(portFolioDataset);
%>
<div class="imui-title">
  <h1>ポートフォリオ サンプル</h1>
</div>
<div class="imui-form-container-narrow">
  <table>
    <tr>
      <td><imchart:chart style="portFolio" data="<%=portFolioInfo %>"></imchart:chart></td>
    </tr>
  </table>
</div>
```

ポートフォリオグラフ用
データ生成部分

グラフ画像表示部分

3.3.3

実行結果



3.4 複合グラフ その1

3.4.1 サンプルで利用するAPI

グラフ描画タグライブラリ	imchart:chart
style 属性	combined

アプリケーション共通モジュール API	jp.co.intra_mart.foundation.chart.graphInfo.BarGraphInfo
	jp.co.intra_mart.foundation.chart.dataset.BarDataset
	jp.co.intra_mart.foundation.chart.data.StandardData
	jp.co.intra_mart.foundation.chart.graphInfo.LineGraphInfo
	jp.co.intra_mart.foundation.chart.dataset.LineDataset

3.4.2 サンプルコード

```
<jsp>
<%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>

<%@ taglib uri="http://www.intra-mart.co.jp/taglib/foundation/chart" prefix="imchart"%>
<%@page import="jp.co.intra_mart.foundation.chart.dataset.*"%>
<%@page import="jp.co.intra_mart.foundation.chart.data.*"%>
<%@page import="jp.co.intra_mart.foundation.chart.graphInfo.*"%>

<%
// 棒グラフサンプルデータ
BarGraphInfo barGraphInfo = new BarGraphInfo();
BarDataset barDataset = new BarDataset();

StandardData barSeries1 = new StandardData();
barSeries1.setName("東京支店");
barSeries1.setValues(new Number[] { 300, 200, 400, 500, 250, 380 });

StandardData barSeries2 = new StandardData();
barSeries2.setName("大阪支店");
barSeries2.setValues(new Number[] { 190, 320, 320, 500, 410, 230 });

barDataset.setCategory(
    new String[] { "1月", "2月", "3月", "4月", "5月", "6月" });
barDataset.addData(barSeries1);
barDataset.addData(barSeries2);
barGraphInfo.setDataset(barDataset);

// 折れ線グラフサンプルデータ
LineGraphInfo lineGraphInfo = new LineGraphInfo();
LineDataset lineDataset = new LineDataset();

StandardData lineSeries1 = new StandardData();
lineSeries1.setName("東京支店");
lineSeries1.setValues(new Number[] { 300, 200, 250, 300, 360, 390 });

StandardData lineSeries2 = new StandardData();
lineSeries2.setName("大阪支店");
lineSeries2.setValues(new Number[] { 240, 300, 350, 180, 380, 340 });

lineDataset.setCategory(
    new String[] { "1月", "2月", "3月", "4月", "5月", "6月" });
lineDataset.addData(lineSeries1);
lineDataset.addData(lineSeries2);
lineGraphInfo.setDataset(lineDataset);
```

棒グラフ用データ生成部分

折れ線グラフ用データ生成部分

```
%>
<div class="imui-title">
  <h1>複合グラフ1 サンプル</h1>
</div>
<div class="imui-form-container-narrow">
  <table>
    <tr>
      <td><imchart:chart style="combined" data="<%=barGraphInfo %>" subData="<%=lineGraphInfo %>"></imchart:chart></td>
    </tr>
  </table>
</div>
```

グラフ画像表示部分

3.4.3 実行結果



3.5 複合グラフ その2

3.5.1 サンプルで利用するAPI

グラフ描画タグライブラリ	imchart:chart
style 属性	bar

アプリケーション共通モジュール API	jp.co.intra_mart.foundation.chart.graphInfo.LineGraphInfo
	jp.co.intra_mart.foundation.chart.dataset.LineDataset
	jp.co.intra_mart.foundation.chart.data.StandardData
	jp.co.intra_mart.foundation.chart.graphInfo.BarGraphInfo
	jp.co.intra_mart.foundation.chart.dataset.BarDataset

3.5.2 サンプルコード

```
<jsp>
<%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>

<%@ taglib uri="http://www.intra-mart.co.jp/taglib/foundation/chart" prefix="imchart"%>
<%@page import="jp.co.intra_mart.foundation.chart.dataset.*"%>
<%@page import="jp.co.intra_mart.foundation.chart.data.*"%>
<%@page import="jp.co.intra_mart.foundation.chart.graphInfo.*"%>

<%
// 折れ線グラフサンプルデータ
LineGraphInfo lineInfo = new LineGraphInfo();
LineDataset lineData = new LineDataset();
StandardData lineSeries = new StandardData();
lineSeries.setName("営業費");
lineSeries.setValues(
    new Number[] { 280, 130, 100, 190, 70, 80, 130, 110, 390, 383 });

lineData.setCategory(new String[] { "1999", "2000", "2001", "2002",
    "2003", "2004", "2005", "2006", "2007", "2008" });
lineData.addData(lineSeries);
lineInfo.setDataset(lineData);

// 棒グラフサンプルデータ
BarGraphInfo barInfo = new BarGraphInfo();
BarDataset barData = new BarDataset();
StandardData barSeries = new StandardData();
barSeries.setName("東京支店");
barSeries.setValues(
    new Number[] { 3200, 3600, 4000, 4200, 3940,
        4300, 6400, 2320, 3420, 5000 });

barData.setCategory(new String[] { "1999", "2000", "2001", "2002",
    "2003", "2004", "2005", "2006", "2007", "2008" });
barData.addData(barSeries);
barInfo.setDataset(barData);

%>
<div class="imui-title">
    <h1>複合グラフ2 サンプル</h1>
</div>

<div class="imui-form-container-narrow">
    <table>
        <tr>
```

折れ線グラフ用
データ生成部分

棒グラフ用データ生成部分

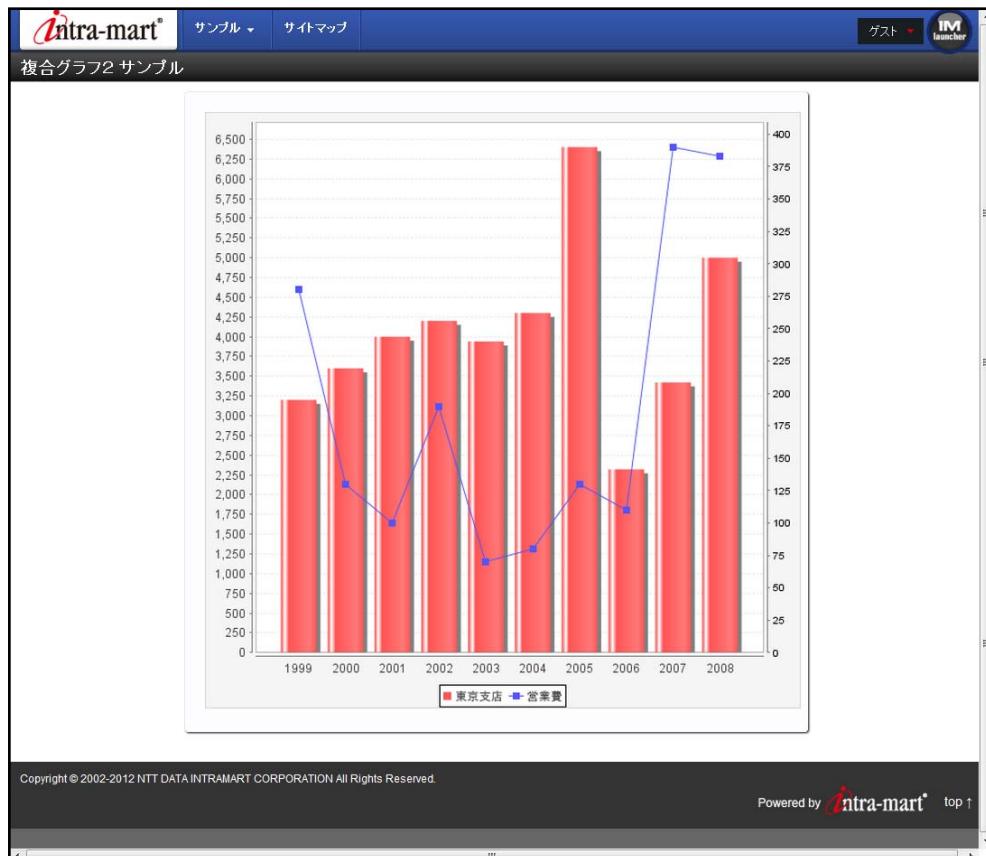
```

<td><imchart:chart style="bar"
    data="<%=>barInfo %>" subData="<%=>lineInfo %>"/></imchart:chart></td>
</tr>
</table>
</div>

```

グラフ画像表示部分

3.5.3 実行結果



4 スクリプト開発モデル

4.1 はじめに

ここでは、スクリプト開発モデルでのグラフ描画モジュールの利用方法(プログラミング)を、簡単なサンプルを例に解説していきます。

4.2 円グラフ

4.2.1 サンプルで利用するAPI

グラフ描画タグライブラリ	chart
style 属性	pie

4.2.2 サンプルコード

```
<html>
  <div class="imui-title">
    <h1>円グラフ サンプル</h1>
  </div>
  <div class="imui-form-container-narrow">
    <table>
      <tr>
        <td><imart type="chart" style="pie" data= pieData /></td>
      </tr>
    </table>
  </div>
</html>

<js>
let pieData = [];

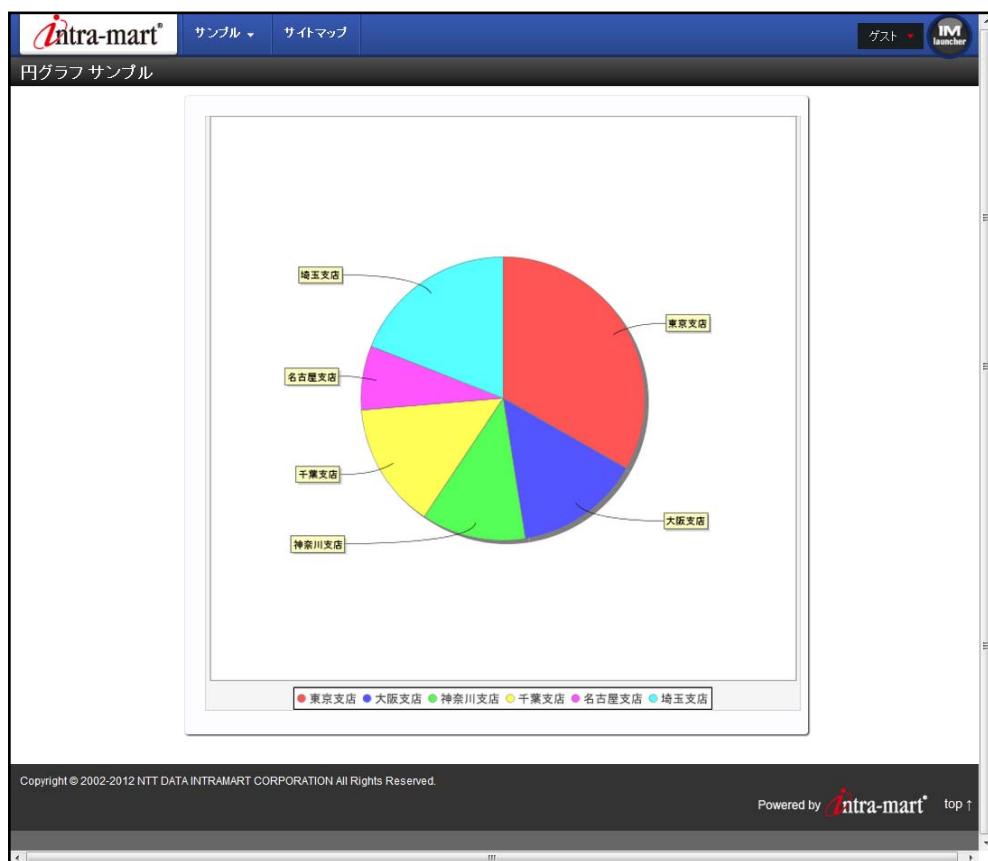
function init(request){

  // 円グラフサンプルデータ
  pieData.dataset = [
    {"name": "東京支店",
     "value": 700
    },
    {"name": "大阪支店",
     "value": 300
    },
    {"name": "神奈川支店",
     "value": 250
    },
    {"name": "千葉支店",
     "value": 302
    },
    {"name": "名古屋支店",
     "value": 155
    },
    {"name": "埼玉支店",
     "value": 400
    }
  ];
}
```

グラフ画像表示部分

円グラフ用データ生成部分

4.2.3 実行結果



4.3 ポートフォリオ

4.3.1 サンプルで利用するAPI

グラフ描画タグライブラリ	chart
style 属性	portFolio

4.3.2 サンプルコード

```
<html>
  <div class="imui-title">
    <h1>ポートフォリオ サンプル</h1>
  </div>
  <div class="imui-form-container-narrow">
    <table>
      <tr>
        <td><imart type="chart" style="portFolio" data=portFolioData /></td>
      </tr>
    </table>
  </div>
```

グラフ画像表示部分

```
<js>
let portFolioData = {};

function init(request){

  // ポートフォリオ・グラフ用サンプルデータ
  portFolioData.xAxis = {
    "max": 100,
    "min": 0,
    "scale": 10
  };
  portFolioData.yAxis = {
    "max": 10000,
    "min": 0,
    "scale": 1000
  };
  portFolioData.dataset = [
    "data": [
      {"name": "東京",
       "value": [
         {"xValue": 15,
          "yValue": 3000,
          "size": 1000
        },
        {"xValue": 90,
          "yValue": 9000,
          "size": 100
        }
      ],
      {"name": "大阪",
       "value": [
         {"xValue": 50,
          "yValue": 5000,
          "size": 3000
        }
      ],
      {"name": "名古屋",
       "value": [
         {"xValue": 35,
          "yValue": 3000,
          "size": 2000
        }
      ]
    ]
  ]
}
```

ポートフォリオグラフ用
データ生成部分

```
    ]]
  ];
}
```

4.3.3 実行結果



4.4 複合グラフ その1

4.4.1 サンプルで利用するAPI

グラフ描画タグライブラリ	chart
style 属性	combined

4.4.2 サンプルコード

```
< html >
  <div class="imui-title">
    <h1>複合グラフ1 サンプル</h1>
  </div>
  <div class="imui-form-container-narrow">
    <table>
      <tr>
        <td><imart type="chart" style="combined" data=lineData subData=barData /></td>
      </tr>
    </table>
  </div>
```

グラフ画像表示部分

```
< js >
let barData = {};
let lineData = {};

function init(request){

  // 棒グラフサンプルデータ(一般棒グラフ)
  barData.dataset = [
    "category": ["1月", "2月", "3月", "4月", "5月", "6月"],
    "data": [
      {
        "name": "東京支店",
        "value": [300, 200, 400, 500, 250, 380]
      },
      {
        "name": "大阪支店",
        "value": [190, 320, 320, 500, 410, 230]
      },
      {
        "name": "千葉支店",
        "value": [210, 260, 300, 320, 330, 360]
      }
    ]
  };

  // 折れ線グラフサンプルデータ
  lineData.dataset = [
    "category": ["1月", "2月", "3月", "4月", "5月", "6月"],
    "data": [
      {
        "name": "東京支店",
        "value": [300, 200, 250, 300, 360, 390]
      },
      {
        "name": "大阪支店",
        "value": [240, 300, 350, 180, 380, 340]
      }
    ]
}
```

棒グラフ用
データ生成部分

折れ線グラフ用
データ生成部分

4.4.3 実行結果



4.5 複合グラフ その2

4.5.1 サンプルで利用するAPI

グラフ描画タグライブラリ	chart
style 属性	bar

4.5.2 サンプルコード

```
< html >
  <div class="imui-title">
    <h1>複合グラフ2 サンプル</h1>
  </div>
  <div class="imui-form-container-narrow">
    <table>
      <tr>
        <td><imart type="chart" style="bar" data=bar subData=line /></td>
      </tr>
    </table>
  </div>

< js >
let line = {};
let bar = {};

function init(request){

  // 折れ線グラフサンプルデータ
  line.dataset = {
    "category": ["1999", "2000", "2001", "2002", "2003", "2004",
                 "2005", "2006", "2007", "2008"],
    "data": [
      {"name": "営業費",
       "value": [280, 130, 100, 190, 70, 80, 130, 110, 390, 383]
      }
    ];
  };

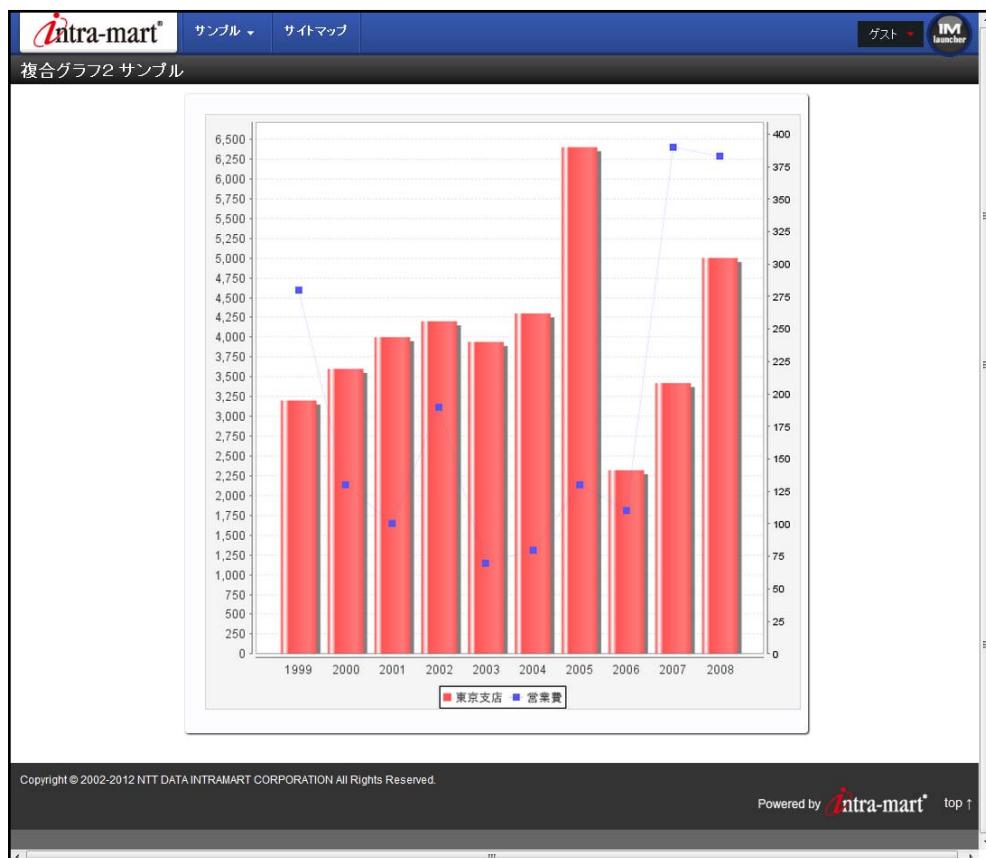
  // 棒グラフサンプルデータ
  bar.dataset = {
    "category": ["1999", "2000", "2001", "2002", "2003", "2004",
                 "2005", "2006", "2007", "2008"],
    "data": [
      {"name": "東京支店",
       "value": [3200, 3600, 4000, 4200, 3940, 4300, 6400, 2320, 3420, 5000]
      }
    ];
}
```

グラフ画像表示部分

折れ線グラフ用
データ生成部分

棒グラフ用
データ生成部

4.5.3 実行結果



5 設定ファイルと設定

5.1 WEB-INF/conf/graph/default.properties

default.properties では、グラフ画像ファイルのタイトルフォントや、背景色などの設定を行います。

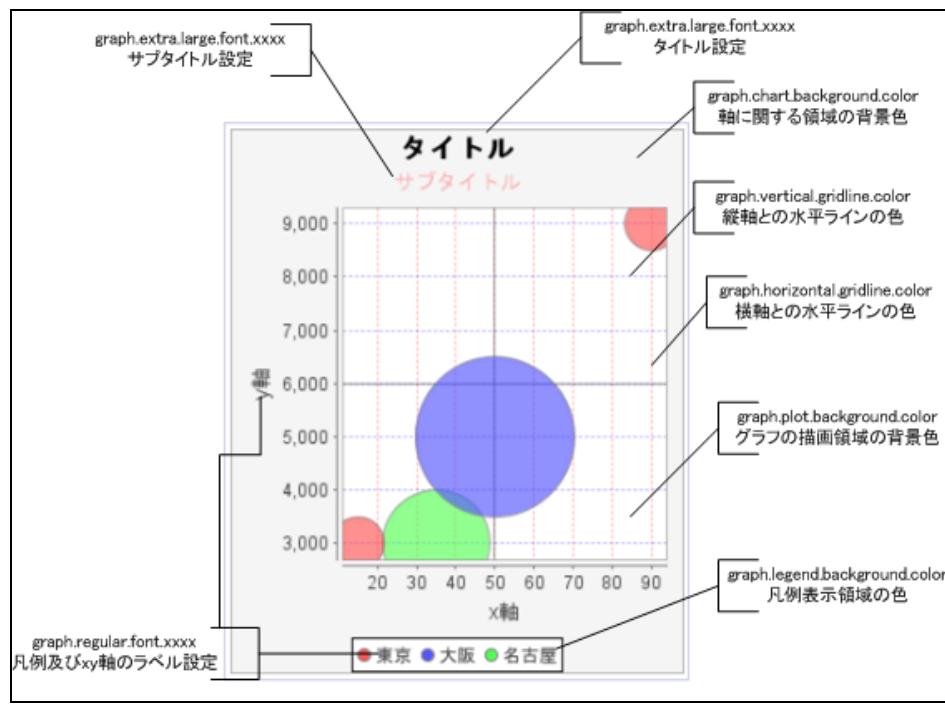
5.1.1 ディレクトリ構造



☆参考☆

この設定ファイルは、JDK の「native2ascii」コマンドにてコード変換を行っております。
編集、参照の際は、「-reverse」オプションでネイティブコードのファイルに変換してください。

5.1.2 設定項目



<<グラフ画像と設定項目>>

5.1.2.1 画像のコンテンツタイプ

パラメータ	説明	デフォルト値
graph.content.type	画像ファイルのコンテンツタイプ 以下のコンテンツ対応が指定可能です。 <ul style="list-style-type: none">• png• jpeg	png

5.1.2.2 画像ファイルのタイトル

パラメータ	説明	デフォルト値
graph.extra.large.font.name	フォント名	dialog
graph.extra.large.font.style	フォントのスタイル 以下のスタイルが指定可能です。 <ul style="list-style-type: none">• BOLD• PLAIN• ITALIC• BOLDITALIC	BOLD
graph.extra.large.font.size	フォントサイズ 十進数で表したポイントサイズで指定します。	18

5.1.2.3 画像ファイルのサブタイトル

パラメータ	説明	デフォルト値
graph.large.font.name	フォント名	dialog
graph.large.font.style	フォントのスタイル 以下のスタイルが指定可能です。 <ul style="list-style-type: none">• BOLD• PLAIN• ITALIC• BOLDITALIC	PLAIN
graph.large.font.size	フォントサイズ 十進数で表したポイントサイズで指定します。	14

5.1.2.4

x軸ラベル、y軸ラベル、凡例

パラメータ	説明	デフォルト値
graph.regular.font.name	フォント名	dialog
graph.regular.font.style	フォントのスタイル 以下のスタイルが指定可能です。 <ul style="list-style-type: none">• BOLD• PLAIN• ITALIC• BOLDITALIC	PLAIN
graph.regular.font.size	フォントサイズ 十進数で表したポイントサイズで指定します。	12

5.1.2.5

画像の中に表示される文字のフォント

パラメータ	説明	デフォルト値
graph.small.font.name	フォント名	dialog
graph.small.font.style	フォントのスタイル 以下のスタイルが指定可能です。 <ul style="list-style-type: none">• BOLD• PLAIN• ITALIC• BOLDITALIC	PLAIN
graph.small.font.size	フォントサイズ 十進数で表したポイントサイズで指定します。	10

5.1.2.6

画像ファイルの高さ

パラメータ	説明	デフォルト値
graph.image.height	画像ファイルの高さ	360

5.1.2.7

画像ファイルの幅

パラメータ	説明	デフォルト値
graph.image.width	画像ファイルの幅	300

5.1.2.8

背景色

パラメータ	説明	デフォルト値
graph.plot.background.color	画像ファイルの背景色	white
graph.plot.outline.color	画像ファイルのグリッド線の背景色	gray
graph.legend.background.color	凡例の背景色	white
graph.chart.background.color	図(グラフ)の背景色	whitesmoke
graph.horizontal.gridline.color	グリッド線(横)の背景色	lightgrey
graph.vertical.gridline.color	グリッド線(縦)の背景色	lightgrey

※ 背景色の指定は、16進数での指定も可能です。

(例) 背景色に白色を指定する場合 → #FFFFFF

5.2 WEB-INF/web.xml

web.xml では、生成した画像ファイルの Storage Service への保存の有無を設定します。

5.2.1 jp.co.intra_mart.foundation.chart.servlet.GraphDrawerServletクラス

5.2.1.1 パラメータ

パラメータ名	説明	デフォルト値
store	生成した画像ファイルの Storage Service への保存の有無を設定します。 true : ファイルを出力する false : ファイルを出力しない	false

5.2.1.2 画像ファイルの保存先

保存先ディレクトリ	% System Storage のルート%/work/im_chart/テナント ID/ログインユーザ名/
ファイル名	15桁のユニーク ID
注意事項	作成された画像ファイルは該当するセッションが破棄されたタイミングで削除されます。

(画像ファイルの保存例)

%System Storage のルート%/work/im_chart/default/aoyagi/5hq5us79nrk7a8t

5.3 テーマ(Theme)機能

グラフ画像ファイルのタイトルフォントや、背景色などの設定を default.properties ではなく、独自のプロパティファイルに定義することが可能となります。

これにより、グラフ画像ファイル毎のタイトルフォントや、背景色の指定を可能とします。

5.3.1 利用手順

- 独自のプロパティファイルを作成して default.properties と同様のディレクトリ (WEB-INF/conf/graph) に配置します。

設定項目は「5.1.2 設定項目」を参考してください。

ここでは、WEB-INF/conf/graph/sample.propertiesを作成するものとします。

- プログラム内で、テーマ(Theme)を指定します。

JavaEE 開発モデル

```
<imart:chart style="portFolio"
    data="<%portletConfig.get("portFolioInfo")%>"
    imageHeight="800"
    imageWidth="600">
</imart:chart>

<imart:chart style="portFolio"
    data="<%portletConfig.get("portFolioInfo")%>"
    theme="sample">
```

プロパティ名称を指定します。

スクリプト開発モデル

```
<imart:chart style="portFolio"  
    data="<%portletConfig.getSetting("portFolioInfo")%>"  
    imageHeight="800"  
    imageWidth="600">  
</imart:chart>
```

```
<imart:chart style="portFolio"  
    data=portFolioData  
    theme="sample">
```

プロパティ名称を指定します。

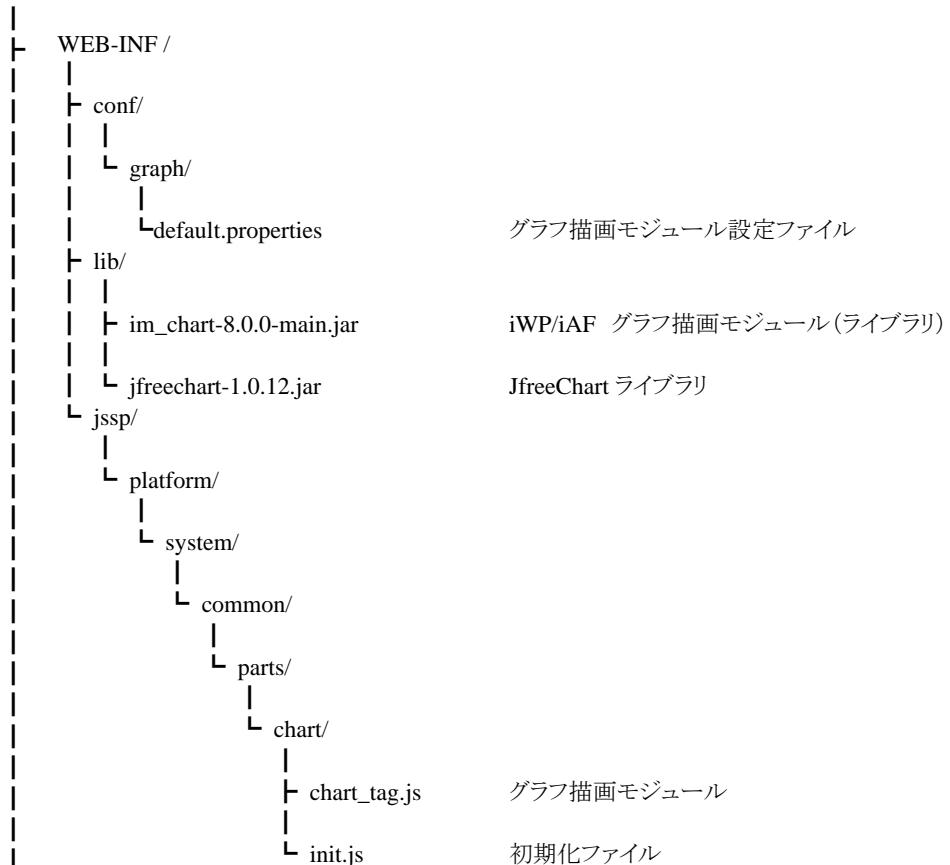
3. 上記手順により、sample.properties で定義した設定内容が指定のグラフ画像ファイルに反映されます。

6 ディレクトリ構造

ここではグラフ描画モジュールに関するライブラリや設定ファイル等のディレクトリについて説明します。
記載するディレクトリ構造は全て、製品インストール時の初期状態となります。

6.1 ディレクトリ構成

WAR ファイル



7 付録

7.1 グラフサンプル

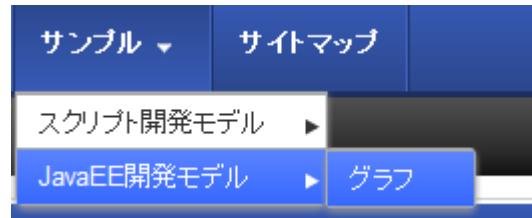
iAP では、グラフ描画モジュールを利用したグラフのサンプルが JavaEE 開発モデル、スクリプト開発モデルそれぞれ提供されています。

グラフのサンプルについては、システム管理者にてサンプルデータインポートを行うことで参照可能となります。

7.1.1 メニュー

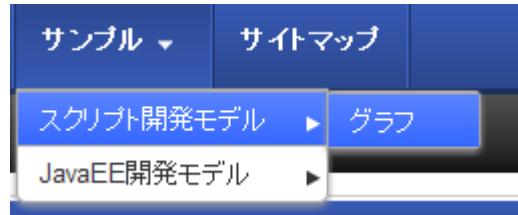
7.1.1.1 JavaEE開発モデル

メニュー:サンプル>スクリプト開発モデル>グラフ



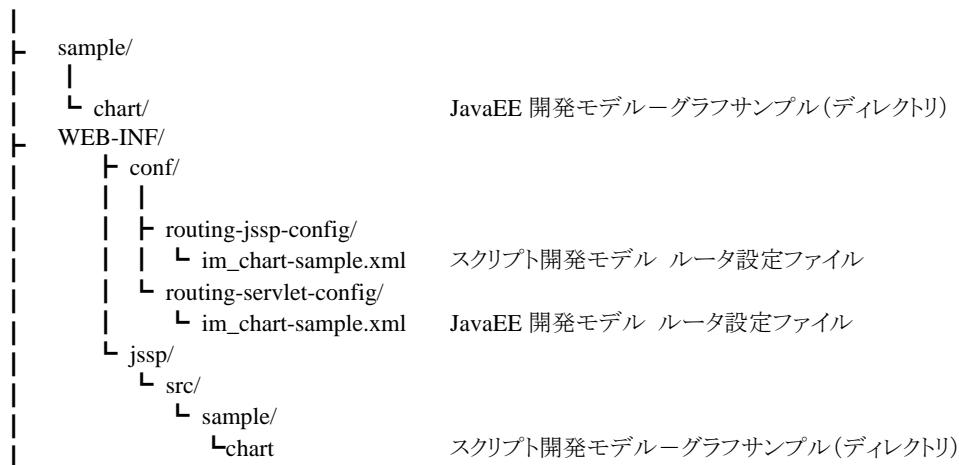
7.1.1.2 スクリプト開発モデル

メニュー:サンプル>JavaEE 開発モデル>グラフ



7.1.2 ディレクトリ構成

7.1.2.1 WARファイル



**intra-mart Accel Platform
グラフ描画モジュールプログラミングガイド**

2012/10/01 初版

Copyright 2000-2012 株式会社 NTT データ イントラマート
All rights Reserved.

TEL: 03-5549-2821
FAX: 03-5549-2816
E-MAIL: info@intra-mart.jp
URL: <http://www.intra-mart.jp/>