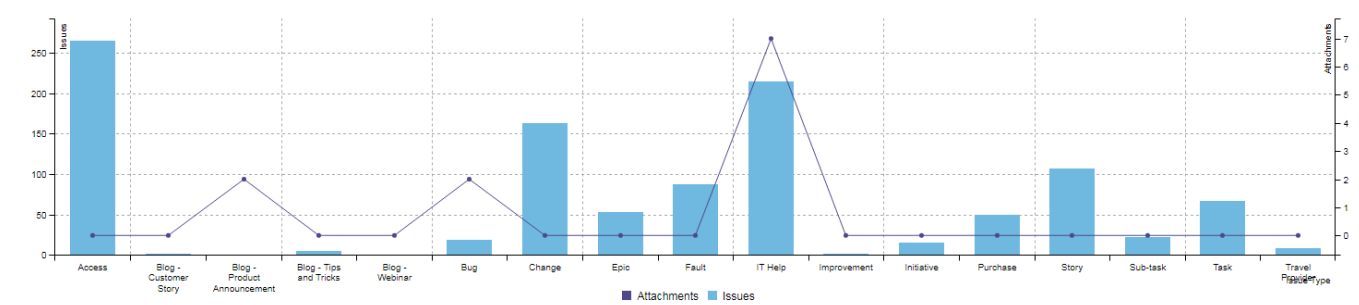


2Y Axes Chart

This chart is an example with 2 Y axes, that can be used for correlating values (Money vs. Time Spent, Original Estimate vs. Time Spent), grouped by a certain category.

[approve Download Scripted Chart Bundle](#)

Chart preview



Parameters

Name		Type	Default
UI	Code		
JQL	JQL	JQL Autocomplete	
Group By X	GroupByX	Group By Picker	Issue Type
Values On Y	ValuesOnY	Value Field Picker	IssueCount
Values On Y2	ValuesOnY2	Value Field Picker	AttachmentCount

Layout Script

```

function formatTooltipAsHours(value, ratio, id, index)
{
    return value.toFixed(2) + ' h';
}

var c3arg = {
    data: chartData,
    grid: {
        y: {
            show: true
        },
        x: {
            show: true
        }
    },
    axis: {
        x: {
            type: 'category',
            label: {
                text: chartData.xtype,
                position: 'outer-right'
            }
        },
        y: {
            label: chartData.custom.yType
        },
        y2: {
            label: chartData.custom.y2Type,
            show: true
        }
    },
    zoom: {
        enabled: true
    }
};

if (chartData.custom && chartData.custom.tooltip)
{
    var tooltipFunction = eval(chartData.custom.tooltip);
    c3arg.tooltip = {
        format: {
            value: tooltipFunction
        }
    };
}

c3.generate(c3arg);

```

Data Script

```

import java.lang.reflect.Field;
import java.math.BigDecimal;
import java.text.DateFormat;
import java.util.Calendar;
import java.math.RoundingMode;

import org.apache.lucene.document.Document;

import com.atlassian.jira.component.ComponentAccessor;
import com.atlassian.jira.user.ApplicationUser;
import com.atlassian.jira.issue.Issue;
import com.atlassian.jira.issue.DocumentIssueImpl;
import com.atlassian.jira.jql.parser.JqlParseException;

```

```

import com.atlassian.jira.jql.parser.JqlQueryParser;
import com.atlassian.query.Query;

import com.decadis.jira.xchart.api.model.Period;
import com.decadis.jira.xchart.api.util.DateUtils;
import com.decadis.jira.xchart.api.model.ChartData;

import com.decadis.jira.xchart.utils.JiraFieldUtil;

BigDecimal transformSecondsToHours(BigDecimal secondsAsBD)
{
    return secondsAsBD.divide(BigDecimal.valueOf(3600.0), 2, RoundingMode.HALF_UP);
}

def addValueAsHours(metaCountGroup, valueExtractor, issue, document, group)
{
    metaCountGroup.addValue(transformSecondsToHours(valueExtractor.get(issue, document)), valueExtractor.
getTitle(), group);
}

def metaCountGroup = chartBuilder.newDataCollector();
def user = ComponentAccessor.getJiraAuthenticationContext().getLoggedInUser();

JqlQueryParser jqlQueryParser = ComponentAccessor.getComponent(JqlQueryParser.class);
Query query = null;
try {
    query = jqlQueryParser.parseQuery(JQL); //JQL is a parameter of type Jql Autocomplete Picker
} catch (JqlParseException e) {
    throw new IllegalArgumentException("Bad JQL: " + query);
}

def valueExtractorY = chartBuilder.getValueExtractor(ValuesOnY);
if ( valueExtractorY == null ) {
    throw new IllegalArgumentException("No value extractor implementation for " + ValuesOnY);
}

def valueExtractorY2 = chartBuilder.getValueExtractor(ValuesOnY2);
if ( valueExtractorY2 == null ) {
    throw new IllegalArgumentException("No value extractor implementation for " + ValuesOnY2);
}

def groupValueExtractor = chartBuilder.getGrouper(GroupByX);

Field documentField;
try
{
    documentField = DocumentIssueImpl.class.getDeclaredField("document");
    documentField.setAccessible(true);
    for ( Issue issue : chartBuilder.getFilterUtils().performSearch(query, user) )
    {
        Document document = (Document) documentField.get(issue);
        for ( String groupX : groupValueExtractor.getGroups((Document) documentField.get(issue)) )
        {
            group = groupValueExtractor.getResolvedValue(groupX, issue);
            if ( JiraFieldUtil.isTimeTrackingField(ValuesOnY) ) {
                addValueAsHours(metaCountGroup, valueExtractorY, issue, document, group);
            }
            else {
                metaCountGroup.addValue(valueExtractorY.get(issue, document), valueExtractorY.getTitle(), group);
            }

            if (JiraFieldUtil.isTimeTrackingField(ValuesOnY2)) {
                addValueAsHours(metaCountGroup, valueExtractorY2, issue, document, group);
            }
            else {
                metaCountGroup.addValue(valueExtractorY2.get(issue, document), valueExtractorY2.getTitle(), group);
            }
        }
    }
} catch (Exception e){
    System.err.println("Exception " + e);
}

```

```

}

def chartData = chartBuilder.newChartData("Issues");

for ( String grpKey : metaCountGroup.keySet() )
{
    chartData.addGroupName(grpKey, groupValueExtractor.getResolvedValue(grpKey, null));
}

chartData.setChartType(valueExtractorY.getTitle(), "bar");
chartData.setChartType(valueExtractorY2.getTitle(), "line");

chartData.setXType(groupValueExtractor.getGroupName());

chartData.setYAxis(valueExtractorY.getTitle(), "y");
chartData.setYAxis(valueExtractorY2.getTitle(), "y2");

chartData.addCustomData("y2Type", valueExtractorY2.getTitle());
chartData.addCustomData("yType", valueExtractorY.getTitle());

if ( JiraFieldUtil.isTimeTrackingField(ValuesOnY) || JiraFieldUtil.isTimeTrackingField(ValuesOnY2) )
{
    chartData.addCustomData("tooltip", "formatTooltipAsHours");
}

chartBuilder.getChartUtil().transformResult(metaCountGroup, chartData, false);

return chartData;

```



Related examples

Title

[Using Jira Software specific classes and Pickers in Scripts](#)

[Story status category grouped by Epic and custom value](#)

[Simple Timeseries Chart](#)

[Simple Table Report](#)

[Simple Scripting Example](#)

[Report - Lucene Group By](#)

[Open issues with average](#)

[Issues in specific status \(Period\)](#)

[Group ordering in scripted charts](#)

[Gantt Diagram](#)

[Customers in a Google Map](#)

[Created vs. resolved with trend](#)

[Comments count by user in JQL result](#)

[Block Search](#)

[2Y Axes Chart](#)

Simple External Database Chart

If you still have questions, feel free to refer to our [support](#) team.