

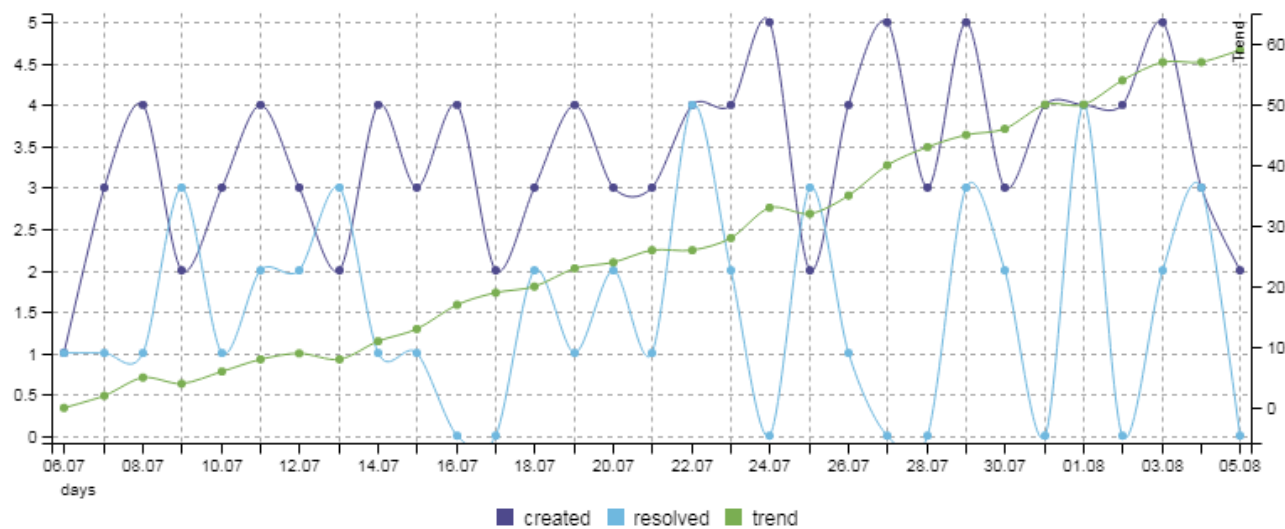
# Created vs. resolved with trend

This chart displays issues created within a specific time period, resolved issues in a specific time period and a trend between these issues.

The time span and the project can be selected via JQL.

[approve](#) [Download Scripted Chart Bundle](#)

## Chart preview



## Parameters

Name	Type	Default
JQL_CREATED	JQL Autocomplete	
JQL_RESOLVED	JQL Autocomplete	
TimePeriod	Time period Picker	DAY
ChartType	Chart Type Picker	spline
Display Created vs. Resolved	BOOLEAN	true
Display Trend	BOOLEAN	true

## Layout Script

### JavaScript Template

```
function formatWeek(d){
  if (d instanceof Date)
  {
    // Copy date so don't modify original
    d = new Date(Date.UTC(d.getFullYear(), d.getMonth(), d.getDate()));
    // Set to nearest Thursday: current date + 4 - current day number
    // Make Sunday's day number 7
  }
}
```

```

        d.setUTCDate(d.getUTCDate() + 4 - (d.getUTCDay()||7));
        // Get first day of year
        var yearStart = new Date(Date.UTC(d.getUTCFullYear(),0,1));
        // Calculate full weeks to nearest Thursday
        var weekNo = Math.ceil(( (d - yearStart) / 86400000) + 1)/7);
        // Return array of year and week number
        return [d.getUTCFullYear(), weekNo];
    }
    return '';
}

function formatQuarter(d)
{
    if (d instanceof Date)
    {
        var q = d.getMonth();
        q = parseInt(q / 3) + 1;
        return 'Q' + q;
    }
    return '';
}

function formatHalfyear(d)
{
    if (d instanceof Date)
    {
        var q = d.getMonth();
        q = parseInt(q / 6) + 1;
        return 'H' + q;
    }
    return '';
}

console.dir(chartData)
c3.generate({
  onrendered: updateFrameHeight,
  data: chartData,
  grid: {
    y: {
      show: true
    },
    x: {
      show: true
    }
  },
  axis: {
    x: {
      type: 'timeseries',
      label: {
        text: chartData.custom.xLabel,
        position: 'outer-left'
      },
      tick: {
        format: eval(chartData.custom.xTickFormat),
        culling: {
          max: 25
        },
        fit: true,
        multiline: false
      }
    },
    y: {
      min: 0
      ,padding: {top:5, bottom:5}
    },
    y2: {
      show: (chartData.colors.trend !== undefined),
      label: 'Trend'
    }
  }
})

```

## Data Script

### Groovy Script

```
import java.lang.reflect.Field;
import java.math.BigDecimal;
import java.util.Date;

import org.apache.lucene.document.Document;

import com.atlassian.jira.component.ComponentAccessor;
import com.atlassian.jira.issue.DocumentIssueImpl;
import com.atlassian.jira.issue.Issue;
import com.atlassian.jira.issue.IssueFieldConstants;
import com.atlassian.jira.issue.fields.CustomField;
import com.atlassian.jira.issue.fields.FieldManager;
import com.atlassian.jira.jql.builder.JqlClauseBuilder;
import com.atlassian.jira.jql.builder.JqlQueryBuilder;
import com.atlassian.jira.jql.parser.JqlQueryParser;
import com.atlassian.query.Query;
import com.atlassian.query.operator.Operator;

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

def fieldManager = ComponentAccessor.getFieldManager();
def i18n = ComponentAccessor.getJiraAuthenticationContext().getI18nHelper();

//helper method to create the JQL Query
def createQuery(String jql, Period period)
{
    JqlQueryParser jqlQueryParser = ComponentAccessor.getComponent(JqlQueryParser.class);
    Query query = null;
    try {
        JqlClauseBuilder jqlClauseBuilder = JqlQueryBuilder.newClauseBuilder(jqlQueryParser.parseQuery(jql));
        query = jqlClauseBuilder.buildQuery();
    } catch (Exception e) {
        throw new IllegalArgumentException("Bad JQL: " + jql);
    }
    return query;
}

Period selectedPeriod = Period.fromString(TimePeriod);

Query query = createQuery(JQL_CREATED, selectedPeriod);
def countGroup = chartBuilder.newDataCollector();

Field documentField;
try
{
    documentField = DocumentIssueImpl.class.getDeclaredField("document");
    documentField.setAccessible(true);

    for ( Issue issue : chartBuilder.getFilterUtils().performSearchOverrideSecurity(query) )
    {
        if(Display_Created_vs_Resolved)
            countGroup.addValue(BigDecimal.ONE, "created", dateUtils.getPeriodGroup(issue.getCreated(),
selectedPeriod));
        if(Display_Trend)
            countGroup.addValue(BigDecimal.ONE, "trend", dateUtils.getPeriodGroup(issue.getCreated(),
selectedPeriod));
    }
} catch (Exception e) {
```

```

    System.out.println("Could not extract groups" + e);
}

query = createQuery(JQL_RESOLVED, selectedPeriod);
try
{
    documentField = DocumentIssueImpl.class.getDeclaredField("document");
    documentField.setAccessible(true);

    for ( Issue issue : chartBuilder.getFilterUtils().performSearchOverrideSecurity(query) )
    {
        if(Display_Created_vs_Resolved)
            countGroup.addValue(BigDecimal.ONE, "resolved", dateUtils.getPeriodGroup(issue.getResolutionDate(),
selectedPeriod));
        if(Display_Trend)
            countGroup.addValue(BigDecimal.valueOf(-1), "trend", dateUtils.getPeriodGroup(issue.
getResolutionDate(), selectedPeriod));
    }
} catch (Exception e) {
    System.out.println("Could not extract groups" + e);
}

def keys = countGroup.getAllKeys()

if(Display_Created_vs_Resolved) {
    countGroup.get("created").fillMissingValues(keys);
    countGroup.get("resolved").fillMissingValues(keys);
}
if(Display_Trend)
countGroup.get("trend").accumulate(keys);

// countGroup.fillMissingValues();
def chartData = chartBuilder.newChartData(il8n.getText("common.concepts.issues"));

if(Display_Created_vs_Resolved) {
    chartData.addGroupName("created", "created");
    chartData.addGroupName("resolved", "resolved");
}
if(Display_Trend) {
    chartData.addGroupName("trend", "trend");
    chartData.setYAxis("trend", "y2");
}

chartData.setxFormat(DateUtils.SimpleDateFormatD3);
chartData.setPeriod(selectedPeriod);
chartData.setType(ChartType);

chartBuilder.getChartUtil().transformResult(countGroup, chartData, false);
return chartData;

```

If you would like to display the date on the x-axis in **MM/DD** format, you need to adjust the function **formatWeek** in the layout script.  
Use the following layout script instead:

## Layout Script

## JavaScript Template

```
function formatWeek(d){
if (d instanceof Date)
{
// Copy date so don't modify original
d = new Date(Date.UTC(d.getFullYear(), d.getMonth(), d.getDate()));
// Set to nearest Thursday: current date + 4 - current day number
// Make Sunday's day number 7

// Original Entry
// d.setUTCDate(d.getUTCDate() + 4 - (d.getUTCDay()|7));
d.setUTCDate(d.getUTCDate() + 1 - (d.getUTCDay()));

// Original Entry
// Get first day of year
// var yearStart = new Date(Date.UTC(d.getUTCFullYear(),0,1));
// Calculate full weeks to nearest Thursday
// var weekNo = Math.ceil(( (d - yearStart) / 86400000) + 1)/7);

// Get month with leading zero
//md = new SimpleDateFormat("MM/dd")
var m = ((d.getMonth() + 1) < 10 ? '0' : '') + (d.getMonth() + 1);
// Get day with leading zero
var dy = (d.getDate() < 10 ? '0' : '') + d.getDate();

// Original Entry
// Return array of year and week number
// return [d.getUTCFullYear(), weekNo];
// Return array of month and first day of week
return [m + '/' + dy];
}
return '';
}
```



## 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.