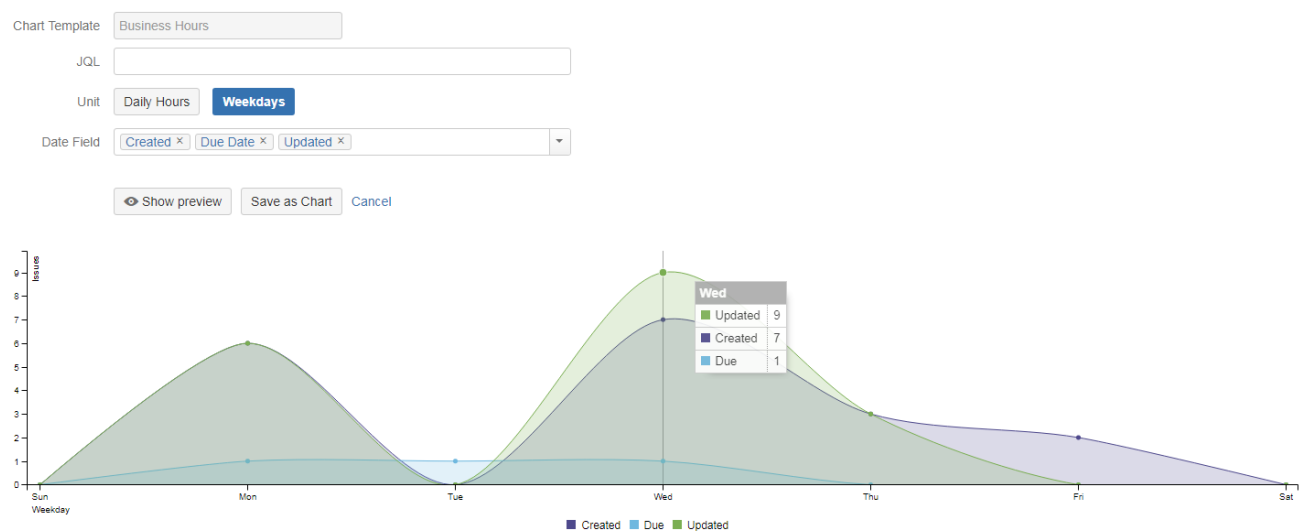


Business hours

The **business hours** chart shows when issues occur and are resolved during working hours.

Since version **1.2.0**, this chart allows configuration.

Chart preview



Parameters

Parameter		Type	Default value
UI	Code		
JQL	jql	JQL Autocomplete	
Unit	Hourly	Boolean	Daily Hours
Date field	DateField	Date field picker (multiple values)	Date of creation

Layout Script

```

c3.generate({
  onrendered: updateFrameHeight,
  data: chartData,
  axis: {
    x: {
      type: 'timeseries',
      label: {
        text: chartData.custom.xname,
        position: 'outer-left'
      },
      tick: {
        format: chartData.custom.xoutput,
        culling: {
          max: 25
        },
        fit: true,
        multiline: false
      }
    },
    y: {
      label: chartData.ytype
    }
  }
})

```

Data Script

```

import java.math.BigDecimal;
import java.util.List;
import java.util.Map;
import java.util.Date;
import java.util.Locale;
import java.text.DateFormat;

import org.apache.lucene.document.Document;

import com.atlassian.jira.issue.Issue;
import com.atlassian.jira.issue.Issue;
import com.atlassian.jira.issue.IssueFieldConstants;
import com.atlassian.query.Query;
import com.atlassian.jira.component.ComponentAccessor;

import com.atlassian.jira.jql.parser.JqlParseException;
import com.atlassian.jira.jql.parser.JqlQueryParser;
import com.atlassian.jira.util.I18nHelper;

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

def i18n = ComponentAccessor.getJiraAuthenticationContext().getI18nHelper();
def data = chartBuilder.newDataCollector();

//Hourly is a Parameter of type Boolean
void updateDataCollectorFor(Date d, String displayString, DateFormat df, dataCollector) {
  if (d != null) {
    if (!Hourly) {
      if (!Hourly) {
        d = dateUtils.getWeekday(d);
      }
      dataCollector.addValue(BigDecimal.ONE, displayString, df.format(d));
    }
  }
}

```

```

JqlQueryParser jqlQueryParser = ComponentAccessor.getComponent(JqlQueryParser.class);
Query query = null;

try {
    query = jqlQueryParser.parseQuery(JQL);
} catch (JqlParseException e) {
    throw new IllegalArgumentException("Bad JQL: " + jql);
}

def df = Hourly ? dateUtils.DAY_TIME_FORMAT : DateUtils.GetSimpleDateFormat();

ForeachDocumentIssue lambda = { Issue issue, Document doc ->
    for (String customDateField: DateField)
    {
        def dateValueExtractor = customDateField != null ? chartBuilder.getValueExtractor(customDateField) :
null;
        boolean isCustomField = customDateField != null && customDateField.startsWith("customfield_");

        Date d = null;
        switch (customDateField)
        {
            case IssueFieldConstants.CREATED:
                d = new Date(issue.getCreated().getTime());
                updateDataCollectorFor(d, il8n.getText("xchart.issue.created"), df, data);
                break;
            case IssueFieldConstants.RESOLUTION_DATE:
                d = issue.getResolutionDate();
                updateDataCollectorFor(d, il8n.getText("xchart.issue.resolved"), df, data);
                break;
            case IssueFieldConstants.UPDATED:
                d = issue.getUpdated();
                updateDataCollectorFor(d, il8n.getText("xchart.issue.updated"), df, data);
                break;
            case IssueFieldConstants.DUE_DATE:
                d = issue.getDueDate();
                updateDataCollectorFor(d, il8n.getText("xchart.issue.duedate"), df, data);
                break;
            default:
                break;
        }
        if (dateValueExtractor != null) {
            if (isCustomField) {
                d = getDate(dateValueExtractor.getValue(issue, doc), locale);
            }
            updateDataCollectorFor(d, dateValueExtractor.getTitle(), df, data);
        }
    }
}

chartBuilder.getFilterUtils().blockSearch(lambda, query, user);

if (Hourly) {
    Set < String > dailyHours = new TreeSet < > ();

    for (int i = 0; i < 24; i++) {
        dailyHours.add(String.format("%02d:00", i));
    }
    data.fillMissingValues(dailyHours);
} else {
    Set < String > days = new TreeSet < > ();

    int fdotw = dateUtils.getFirstDayOfWeek();

    for (int i = 0; i < 7; i++) {
        Date d = dateUtils.getWeekday(fdotw);
        days.add(df.format(d));
        fdotw++;
        if (fdotw > Calendar.SATURDAY) {
            fdotw = Calendar.SUNDAY;
        }
    }
}

```

```
    }  
    data.fillMissingValues(days);  
}  
  
ChartData chartData = chartBuilder.newChartData(i18n.getText("common.concepts.issues"));  
chartData.setxFormat(Hourly ? "%H:%m" : DateUtils.SimpleDateFormatD3);  
chartData.setType("area-spline");  
chartData.addCustomData("xname", Hourly ? i18n.getText("xchart.time") : i18n.getText("xchart.weekday"));  
chartData.addCustomData("xoutput", Hourly ? "%H:%M" : "%a");  
  
chartBuilder.getChartUtil().transformResult(data, chartData);  
return chartData;
```

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