# Subreports

## ReportMill Subreports User Guide

Sometimes it is useful for a report to have an element that references and effectively embeds another template. This is accomplished easily using the Subreport component.

Note that many reporting tools need such a feature to nest tables, but ReportMill accomplishes that easily with Table Groups, without the need for separate external templates (though Subreports are an option for nested tables as well).
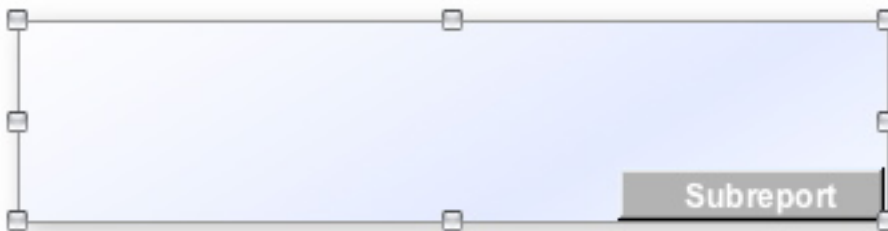
## Table of Contents

# ★ Subreports

## Overview

Subreports are a powerful feature that let you import a separate external template into a master template. This can be useful for situations where multiple templates wish to include the same header or footer. The header or footer can be designed as a separate template, then referenced by other templates.
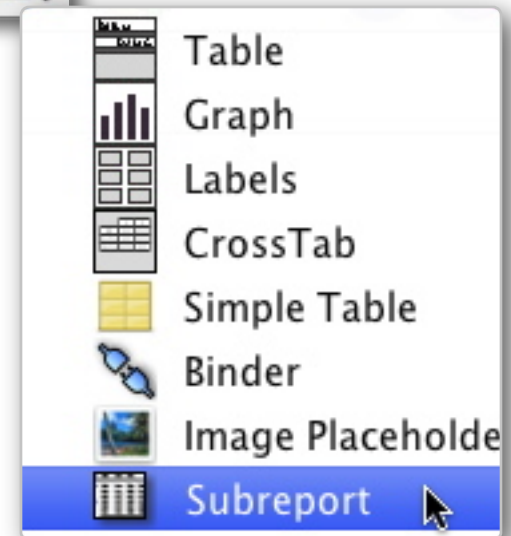
## Adding a Subreport

To add a subreport, simply click the Add Shapes button in the toolbar and select the Subreport menu item. This will add an empty subreport template to the document. Simply position and size this to cover the region of the page that you want to be filled with an imported document.

**Add Shapes Button
in ToolBar**

Table
Graph
Labels
CrossTab
Simple Table
Binder
Image Placeholde
Subreport
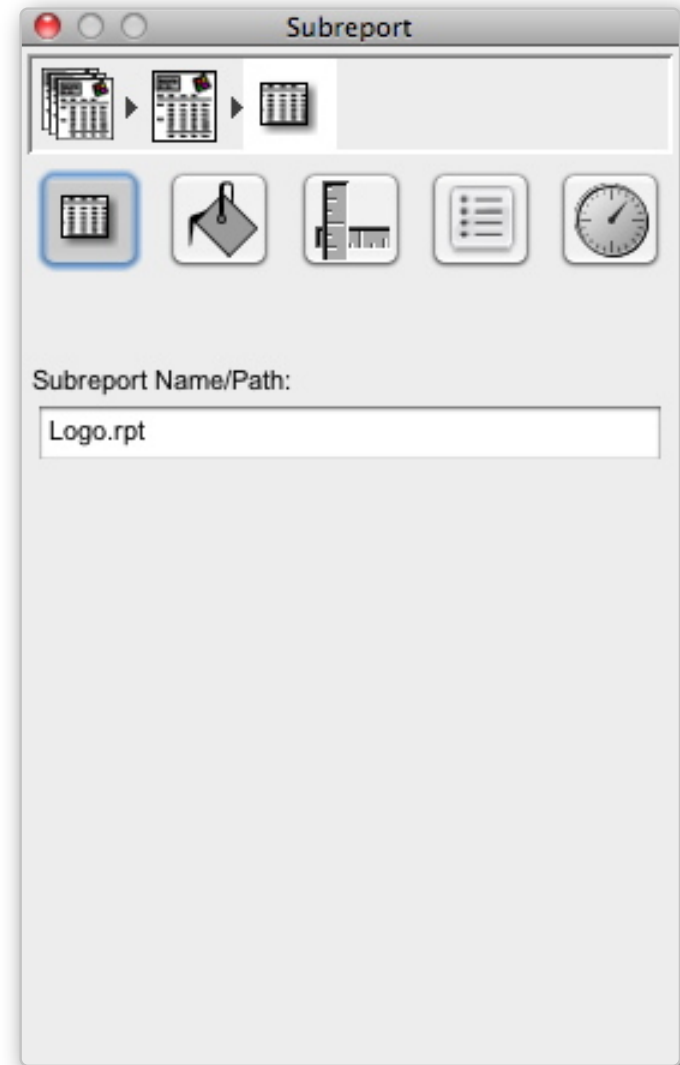
**Add Shapes Menu**

Subreport

**Subreport Template Component**

# Subreports

## Configuring a Subreport

To configure a subreport, simply enter the name of a template, ideally one that has been saved in the same directory as the current template. The subreport name can also be a path.

Once a subreport component has been configured with a template, it will be painted with the contents of that template. Again it can be sized and positioned.

# Subreports

## Resolving Referenced Subreports

By default, ReportMill documents know how to resolve templates that are found in the filesystem. If instead your templates are loaded from inputstreams or other more obscure sources, you can override the RMDocument getSubreport method to add support for any template retrieval mechanism.

```java
// Load template from File, String path, input stream, etc.
RMDocument template = new RMDocument(aSource) {

    /**
     * Override for custom subreport resolution.
     */
    public RMDocument getSubreport(String aName)
    {
        return new RMDocument(getInputStream(aName));
    }
}
```