

Overview

[Overview](#) / [Product overview](#) / [Features](#) / [No-charge product enhancement features](#) / [Reports](#) / Usage scenario for Reports

Usage scenario for Reports

In this scenario, a printing company wants to generate daily reports that show the pages printed on each printer and the jobs printed by each customer. They want to create a reporting dashboard with the two reports and publish the dashboard to a server that company management can access.

The printing company installs the **Reports** feature on their RICOH ProcessDirector for Windows primary server, which includes the PostgreSQL database program. Then, they open the **Administration** page and select **Reports ⇒ Database Settings** to create the reports database.

The printing company wants to generate two reports: **Pages printed by printer** and **Jobs printed by customer**. They can collect that data for both of those reports using the **Job Print Progress** data collector. For the first report, they need to capture the Pages stacked and Previous Printer job properties; for the second report, they need to capture the Job number and the Customer name properties.

The user selects **Reports ⇒ Data Collectors** and opens the **Job Print Progress** data collector. On the data collector property notebook, the user finds the **Job properties to capture** list and selects these properties:

- Pages stacked
- Previous Printer
- Job number
- Customer name

The user names the table `job_printing` and enables data collection for the data collector.

After the user clicks **OK**, RICOH ProcessDirector captures the values of the properties in the PostgreSQL database each time that a job starts or stops printing.

After enabling data collection for the `job_printing` table and populating it with data, the printing company installs the Tableau business intelligence (BI) tool on a remote server. They configure RICOH ProcessDirector to let Tableau connect to the PostgreSQL database remotely.

Using Tableau, the printing company connects to the `job_printing` table in the PostgreSQL database. They choose to update the data in Tableau automatically when the data is captured in PostgreSQL.

To prepare the **Pages printed by printer** report, the printing company:

- Creates a worksheet named **Pages printed by printer**.
- Populates the worksheet with the values of the **Job.PreviousPrinter** and **Job.PagesStacked** properties.
- Renames **Job.PreviousPrinter** to **Printer** and **Job.PagesStacked** to **Pages**.
- Chooses a pie chart to visualize the data.

When the printing company finishes preparing the report, they save the worksheet as a Tableau workbook (TWB) file.

To set up the **Jobs printed by customer** report, the printing company:

- Creates a worksheet named **Jobs printed by customer**.
- Populates the worksheet with the values of the **Job.ID** and **Job.CustomerName** properties.
- Renames **Job.ID** to **Job** and **Job.CustomerName** to **Customer**.
- Chooses a bar chart to visualize the data.

When the printing company finishes preparing the report, they save the worksheet as a Tableau workbook (TWB) file.

The printing company creates a reporting dashboard with the two reports and saves the dashboard as a Tableau workbook (TWB) file. The workbook is then published to a server that company management can access to display the dashboard.

Parent topic: [Reports](#)