

Scan2Print: Printing labels by scanning

Scan2Print is a dedicated, mobile-optimized page for printing labels and documents straight from a barcode scan. It is built for handheld terminals (MDE) and smartphones, but works just as well on a desktop workstation with a USB scanner. Configure the station once, then every scan becomes a print job, no forms, no clicks, no PDF preview in between.

Typical use cases are inbound goods labels, article and batch labels, inspection tags or pallet labels, anywhere a worker scans a value and needs a printed label a second later.

Screenshot placeholder: `scan2print.index-mobile.png`

Scan2Print page on a handheld (approx. 360 px width): top bar with configuration summary and gear button, two scan fields, Print button.

How It Works

1. Open **Scan2Print** from the main navigation.
2. On the first visit, a configuration dialog opens. Select the report, the printer, the number of copies and fill in the fixed values for this station.
3. The page now shows one input field per scan parameter of the report, stacked on top of each other.
4. Scan the values. Every scan ends with **Enter**, which moves the cursor to the next field. The Enter on the last field triggers the render.
5. VeloxFactory renders the report, creates a print task and returns to the empty scan form. The cursor is back in the first field, ready for the next scan.

Under the hood, Scan2Print uses the same rendering pipeline as the [Generate PDF page](#) and the API. Every Scan2Print submit is a regular render request with a print task attached, so everything you know about print tasks, history records and SQL data adapters applies here as well.

Permissions

Scan2Print has its own permission, found in the user settings under the section **Scan2Print**:

Permission	Purpose
<code>scan2print:use</code>	Shows the Scan2Print navigation item and grants access to the page. Administrators (<code>global:admin</code>) have access automatically.

Screenshot placeholder: `scan2print.user-permission.png`
User edit page, section *Scan2Print* with the *Use* checkbox.

`scan2print:use` is all a scan user needs. Every Scan2Print submit runs through the regular rendering pipeline, which checks its own permissions for rendering, print tasks and history records. The Scan2Print permission therefore implicitly includes these permissions, there is no need to grant them separately:

Implicit permission	Required for
Report Configs: Read	Selecting a report and rendering it.
Report Print Tasks: Read and Create	Creating the print job after every scan and checking the print queue.
Report History Records: Create and Update	Storing the render in the report history, if enabled in the station configuration.

As a consequence, a Scan2Print user also sees the **Reports** and **Print Queue** items in the navigation and can open report configurations read-only, including their Generate PDF page.

i Dedicated scan users land on Scan2Print directly. Users who hold `scan2print:use` but may not create or edit report configurations are redirected to Scan2Print when they open the application root. Bookmarking the Scan2Print page on the device works as well, after a login VeloxFactory returns to it automatically.

Preparing a Report for Scan2Print

Scan2Print does not need a special report type. It works with any report configuration and decides by **parameter name prefix** which values are scanned and which are fixed:

Prefix	Role	Filled in
P_SCAN_	Scan field. Appears as an input on the scan page. Always mandatory.	On every scan
P_STATIC_	Fixed value for the station, e.g. client name, storage location or inspector.	Once, in the configuration dialog
P_RESOURCE_	Image resource, handled automatically as usual.	Never
any other	Regular parameter. Not filled by Scan2Print, but can be delivered by the SQL query (see below).	Never

The scan fields appear in the order in which the parameters are declared in the `.jrxml`. The field label is derived from the parameter name, `P_SCAN_ARTICLE_NUMBER` is shown as **Article Number**. Scan and static fields use the same typed inputs as the Generate PDF page, a `java.sql.Date` static parameter becomes a date picker, a `java.lang.Integer` scan field a numeric input.

A minimal parameter block looks like this:

```
<parameter          name="P_SCAN_ARTICLE_NUMBER"          class="java.lang.String">
                                <property>

</parameter>
<parameter          name="P_SCAN_ARTICLE_BATCH"           class="java.lang.String">
                                <property>

</parameter>
<parameter          name="P_STATIC_INBOUND_DATE"          class="java.lang.String">
                                <property>

</parameter>
```

Which reports can be selected

Only reports that Scan2Print can actually fill are offered in the configuration dialog. A report is selectable if **all** of the following apply:

- It has at least one `P_SCAN_` parameter.
- None of its `P_SCAN_` parameters is of type `java.lang.Boolean`, a toggle cannot be scanned.

- It has **no detail band**, or it has an **SQL data adapter**. Scan2Print never sends report lines, so detail data can only come from a query.
- It has no **required** parameter outside `P_SCAN_` and `P_STATIC_`, because Scan2Print has no way to fill it.

If your report does not show up in the dropdown, check it against this list first.

Enriching the label with SQL

The real strength of Scan2Print comes with an SQL data adapter. Scan only the key, for example an article number, and let the query fetch everything else. Scan values are available in the query like any other parameter, and every column aliased with a parameter name is transferred into that parameter:

```
select
    , b.barcode
    , a.supplier
from
    article a
inner join articlebarcode b on a.id = b.articleId
where a.articleNumber = :P_SCAN_ARTICLE_NUMBER
```

Parameters that are filled by the query must be marked as **not required** in the `.jrxml`. VeloxFactory validates required parameters before the query runs, so a required parameter that only the query delivers would always fail.

i A query must return at least one row. If a report has an SQL data adapter and the query finds nothing, for example because an unknown article number was scanned, the render fails with *"No data delivered (or fetched via SQL using parameters)..."*. No print task is created, so no half-empty label is printed. If the query returns several rows, the values of the last row are used for the parameters.

Configuring the Scan Station

The configuration dialog opens automatically when no configuration exists yet. Afterwards, it can be reopened at any time with the **gear** button in the top bar of the page, which also shows a compact summary of the active configuration (report, printer, copies, broadcast ID and static values).

Option	Default	Description
Report	-	Searchable list of all Scan2Print capable reports. Required.
Printer Name	-	The target printer for every print task of this station. Required.
Copies	1	Number of copies per scan, between 1 and 100.
Broadcast ID	-	Optional WebSocket channel ID for real-time notification of the print service. Leave empty to rely on polling.
Create History Record	On	Stores every scan render in the report history, including request, response and PDF. Can be switched off for high-volume stations.
Static Parameters	-	One typed input per <code>P_STATIC_</code> parameter of the selected report. They are loaded as soon as a report is selected and replaced when you switch to another report. All of them are required, except boolean toggles.

Screenshot placeholder: `scan2print.config-modal.png`

Configuration dialog with selected report, printer, copies, broadcast ID, history toggle and loaded static parameters.

Click **Save** to store the configuration. If something is missing or invalid, VeloxFactory shows an error message and reopens the dialog with your input still in place.

To reset the station, open the dialog and click **Clear configuration**. The scan form disappears and the dialog asks for a new configuration.

i Static values are fixed. A static date stays exactly as entered, it does not move on to the next day. If a label should always show the current date, use a default expression in the `.jrxml` instead of a static parameter.

Scanning and Printing

With a valid configuration, the page shows all scan fields of the report, large and full width, with the cursor already in the first field.

- **Enter** in a filled field moves the cursor to the next field and selects its content.
- **Enter** in the last field submits the form and starts the render.
- **Enter** in an empty field does nothing, the cursor stays where it is. All scan fields are mandatory.
- The **Print** button below the fields submits the form as well, which is handy for manual input on a desktop.

Screenshot placeholder: `scan2print.scan-form.png`

Scan form on desktop with a configured report, cursor in the first scan field.

After submitting, the form is locked until the page has reloaded. Scanners that send their data in quick bursts or add an extra Enter therefore never create duplicate print jobs. Reloading the page after a print does not print again either.

The Result

Success

A short green notification confirms the print job for two seconds, including report name, printer and number of copies. The scan fields are empty again and the cursor is back in the first field. The print task is now in the print queue and will be picked up by the print service.

Screenshot placeholder: `scan2print.result-success.png`

Green notification after a successful scan, empty scan fields.

Errors

If the render fails, a red block appears in the page body. It stays visible until the next scan and contains:

- all error messages returned by VeloxFactory, e.g. an unknown article number or a missing value,
- the values that were scanned, so the worker knows which item failed,

- a link to the print queue.

The scan fields are cleared and the cursor is back in the first field, so the worker can simply scan again.

Screenshot placeholder: `scan2print.result-error.png`

Red error block with error message (e.g. unknown article number), scanned values and link to the print queue.

i Check the print queue before rescanning after a WebSocket error. If the broadcast to the print service fails, for example because Reverb is not running, the print task has already been created and is still printed by polling. Scan2Print shows an error in this case, a rescan would print the label twice.

Where the Configuration Is Stored

The station configuration is stored in the **user session** on the server. This keeps each device independent, two handhelds logged in with different users can serve different reports and printers at the same time.

- The configuration survives page reloads and navigation within VeloxFactory.
- It is removed on logout and when the session expires after the configured idle time (`SESSION_LIFETIME`). The worker then logs in again and reconfigures the station. After a session timeout, a pending scan cannot be submitted anymore and the page has to be reloaded.
- Static values are stored by parameter name. If the report is deleted, loses its Scan2Print capability, or a new `.jrxml` with additional required static parameters is uploaded, the configuration is reset automatically on the next page load, a warning is shown and the configuration dialog opens.

Scanner Requirements and Tips

- Scan2Print expects a scanner in **keyboard mode** (keyboard wedge), as used by nearly all handheld terminals and USB scanners.

- Configure the scanner to send **Enter** as suffix. A **Tab** suffix only moves the cursor and never submits the form.
- Camera scanning is not part of Scan2Print. Smartphones need a hardware scanner or a scanner app that types into the focused field.
- On smartphones, the on-screen keyboard opens when a field is focused. This is intended, it keeps manual input possible.
- For barcodes on the printed label, prefer **Code128** when the label should reproduce the scanned value exactly. EAN-13 components pad shorter values such as EAN-8 codes with zeros and recalculate the check digit, so the printed barcode would differ from the scanned one.

Revision #6

Created 2026-09-16 20:17:15 UTC by Benjamin Fischer

Updated 2026-09-20 19:31:05 UTC by Benjamin Fischer