



ABAP Package V4 Release Notes

Rev-2024-03

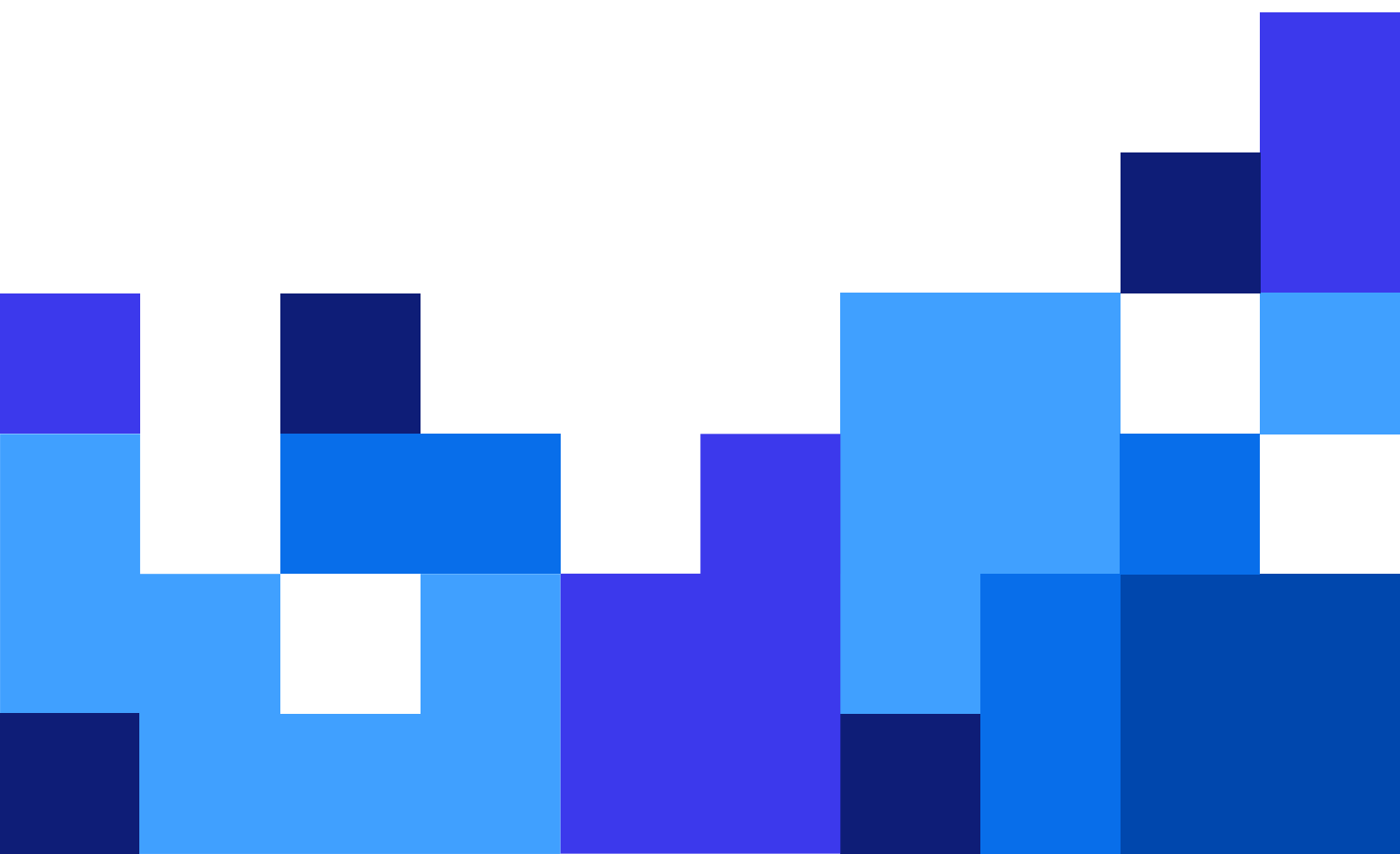


Table of Contents

1. Core ABAP Package	3
1.1. Integration scenario through Cloud Platform Integration (CPI)	3
1.2. Support for RFC destination (HTTP REST) when using integration with PI/PO	3
1.3. New method: EXPORT DATA	4
1.4. Support for printers from multiple Automation endpoints	4
1.5. Placeholders for custom methods	5
1.6. Processing custom data from Automation feedback messages	5
1.7. Support for label copies	6
1.8. Logging IDs of used Automation servers	6
1.9. Adding SP01 events with standard table update modules	6
1.10. Custom titles for SAP spooler event logging	7
1.11. "Location" attribute is available in the printer list (version 4.2)	7
1.12. Updates based on the SAP Code Inspector findings (Version 4.2)	8
1.13. Support for printer's paper tray (version 4.5)	8
1.14. Support for changing label settings at print-time (version 4.5)	8
1.15. The spool requests in SAP are linked with your output devices (version 4.5)	9
2. Demo transaction	10
2.1. Filtering printer lists by selected Automation server	10
2.2. EXPORT_DATA field available in the UI	10
2.3. Clearing previews when selecting new label templates	10
3. Automation configuration	11
3.1. Support for Loftware Document API	11
3.2. New placeholder support for inbound and outbound messages (updated in 4.5)	11
3.3. Support for label copies	11
3.4. Inbound XML message samples	12
3.5. Sending label preview to a list of email recipients (version 4.5)	12
4. Bugfixes	13
4.1. Demo transaction can't access and display Loftware logo	13
4.2. Applying the value of _PATH_DATAFILES parameter (Version 4.2)	13
4.3. Write to SAP spool fails from your own class (version 4.5)	13
4.4. The error message is raised when no Data element is provided for the PRINT, PREVIEW, or PRINT_JOB methods (version 4.5)	14
5. Important upgrade notices	15

1. Core ABAP Package

1.1. Integration scenario through Cloud Platform Integration (CPI)

SAP **Cloud Platform Integration (CPI)** is cloud middleware software. While SAP Process Orchestration (PO) is an on-premise middleware solution, CPI is a cloud-based integration platform that allows data synchronization between multiple systems.

CPI helps you connect cloud and on-premise applications with other SAP and non-SAP cloud and on-premise applications. CPI can process messages in real-time for different companies, organizations, or departments within one organization.

The Software ABAP Package supports communication through CPI, which acts as a central hub for all inbound and outbound messages for SAP systems. The ABAP Package sends messages to CPI which routes them to Software Automation.

1.2. Support for RFC destination (HTTP REST) when using integration with PI/PO

SAP PI/PO (Process Integration/Process Orchestration) is an integration platform that seamlessly integrates SAP and non-SAP applications (like Software Automation) in your organization. You use PI/PO in complex environments when you need tight control over communication between systems and only allow minimum communication. All messages go through a central point (with PI/PO systems acting as brokers between clients) without rogue point-to-point connections.

The ABAP Package previously supported Web Service calls to PI/PO.

The ABAP Package update supports HTTP REST endpoints for PI/PO configuration. You can select RFC destinations for Automation endpoints where you deploy the provided Automation configuration (and the contained Cloud trigger).



NOTE

RFC destinations now allow cloud-to-cloud connectivity scenarios between SAP and Software.

1.3. New method: EXPORT DATA

Many SAP integration projects use Software software to get data from SAP to intermediate SQL databases (on Cloud or on-premise SQL). You can customize Software Form applications to read data from intermediate databases and print the data on labels.



NOTE

The ABAP Package can act as a data extractor and push data out of SAP. The print process then initiates outside of SAP GUI/Fiori in Software applications.

Previously, customers and Software PSG (Professional Service Group) often modified Automation configurations to export data. This approach required reconfiguring existing Automation configuration methods, like updating the PRINT method to save data in SQL servers instead of printing labels.

Now, the ABAP Package includes a built-in data export method. XML Header sections contain a new **EXPORT_DATA** element you can use as a Boolean flag to signal Automation to save received data. Automation configurations contain placeholders for user-defined actions. The data-saving part is customer-specific and customizable for your specific requirements.



NOTE

Software Automation receives data from SAP, but you must define how to save the data you receive.

1.4. Support for printers from multiple Automation endpoints

You could previously use multiple Automation servers to print labels, but you had to manually keep records of which printers were available from which server.

The ABAP Package update provides native support for multiple Automation endpoints. Now when you query the Automation server for a list of printers, the ABAP Package associates all printers with the Automation server that provides the printer list. The expanded internal printer table now includes an additional field for storing the Automation server.



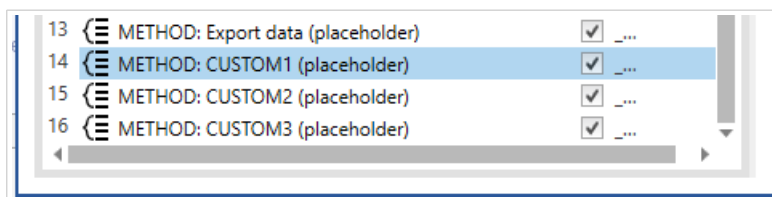
NOTE

The ABAP Package stores printer names and Automation server names so you can filter printers by Automation server.

1.5. Placeholders for custom methods

The ABAP Package controls Automation backend execution by sending XML messages containing Boolean directives to run specific methods like “print a label” or “return a label preview”. The XML message structure is locked inside the ABAP Package and is not user-configurable. Some customers wanted to use the messaging mechanism provided with the ABAP Package to run tasks not originally envisioned by Software when developing the ABAP Package.

This update expands the XML structure header with placeholders (Boolean directives) you can use to run a specific set of custom actions you develop in Automation. The provided Automation configuration contains placeholders for these custom actions, but it’s your responsibility to configure the actions in Automation. You can configure up to 3 custom methods.



The Automation configuration placeholders where you can define your custom set of actions.



NOTE

For example, when you set the field **CUSTOM1** to **True**, Automation executes all actions defined inside the **CUSTOM1** placeholder in the Automation configuration.

1.6. Processing custom data from Automation feedback messages

The XML message responses from Automation now contain new elements. The new elements provide a transport mechanism for sending custom data from Automation to SAP. These are custom user fields that Software Automation does not use internally.

```
<?xml version="1.0"?>
<Feedback>
  <ResponseText>[ResponseText]</ResponseText>
  <PrinterStatus>[PrinterStatusFeedback]</PrinterStatus>
  <Preview>[Preview64]</Preview>
  <PrinterList>[PrinterList]</PrinterList>
  <LabelVariables>[LabelVariables]</LabelVariables>
  <PrintJob>[Job64]</PrintJob>
  <LabelCatalog>[LabelCatalog]</LabelCatalog>
  <Custom1></Custom1>
  <Custom2></Custom2>
  <Custom3></Custom3>
</Feedback>
```



NOTE

Custom fields are not associated with any variables in the provided Automation configuration and are sent back empty. For example, to send back the value of the variable **MyVariable** in the custom field **Custom1**, update the XML response like this:

```
<Custom1>[MyVariable]</Custom1>
```

The ABAP Package parses the values of these new elements and provides them to the SAP application that initiates the API call to the ABAP Package.

1.7. Support for label copies

You can control the printing of label copies from SAP and specify the number of label duplicates for Software Automation. When your SAP provides all data for label printing, you don't typically need a setting for label copies. However, when SAP provides only partial data and Automation has to enrich it with data from an external source, label copies may be useful.

The XML payload header now contains the additional element **QUANTITY_IDENTICALCOPIES**. You can define a global number of copies that apply to all labels in the provided data. You can also individually define the number of label copies for each label in a job.

1.8. Logging IDs of used Automation servers

The ABAP Package can send requests to multiple Automation servers. If a company with production on multiple sites runs Automation on each site, Software Automation logs details of each print request in Control Center, including Automation server names.

You can also configure the ABAP Package to log a subset of data in the SAP Spooler (Tcode SP01). When you enable SP01 logging, the destination name of the Automation server you select is saved with the rest of the data to ensure tracing and help with debugging.



NOTE

Software Automation server names are defined by the logical port names where Automation is accessible.

1.9. Adding SP01 events with standard table update modules

Previously, the ABAP Package used 2 direct updates to the spool requests table (TSP01) to show Automation server responses. The updates set spool items as completed:

- Update for the red/green status field.
- Update for the number of pages.



NOTE

SAP discourages updating SAP tables directly; it's not the best practice.

The updated ABAP Package uses standard SAP spool update functions to update statuses and to remain consistent using a standard functional module or BAPI to ensure tables are properly locked during updates. The ABAP code now copies standard SAP functions to update spoolers with additional custom code for updating red/green status fields and the number of pages.

1.10. Custom titles for SAP spooler event logging

Some customers that use SAP event logging want to link each Loftware print event with the SAP data printed on labels to see document numbers displayed in SAP Spoolers and easily locate printed labels.

Output Controller: List of Spool Requests						
Spool no.	Type	Date	Time	Status	Pages	Title
<input type="checkbox"/> 8211		09.08.2021	17:59	-	2	NiceLabel Request
<input type="checkbox"/> 8210		09.08.2021	17:59	Compl.	1	NiceLabel PDF
<input type="checkbox"/> 8209		09.08.2021	17:58	Compl.	1	NiceLabel Request
<input type="checkbox"/> 8208		09.08.2021	17:58	-	2	NiceLabel Request
<input type="checkbox"/> 8207		09.08.2021	17:58	Compl.	1	NiceLabel PDF
<input type="checkbox"/> 8206		09.08.2021	17:56	Compl.	1	NiceLabel Request
<input type="checkbox"/> 8170		09.08.2021	17:14	Compl.	1	NiceLabel Request
<input type="checkbox"/> 8039		09.08.2021	13:45	-	2	NiceLabel Request

You can customize the Title field for each spool request.

The logging function exposes input parameters for Title fields. The values you provide are used in Title fields.

1.11. "Location" attribute is available in the printer list (version 4.2)

A new attribute "Location" was added to the printer details in the list of available printers. The attribute will get its value from a **Location** field in the Windows printer driver. The printer determination in SAP can be easier if you know about the physical location of your printers.



NOTE

You must use the ABAP Package transport package with the index number 92 or above (available with NiceLabel 10.2 release) to have these changes.

1.12. Updates based on the SAP Code Inspector findings (Version 4.2)

The SAP Code Inspector is a generic tool to check repository objects. It checks performance, security, syntax, and adherence to naming conventions of individual or sets of repository objects. As a result of an inspection, you receive information messages, warning messages, or error messages on different properties of the examined objects.

Based on the findings provided by the tool, we have updated the ABAP Package in terms of security, performance, and robustness.



NOTE

You must use the ABAP Package transport package with the index number 92 or above (available with NiceLabel 10.2 release) to have these changes.

1.13. Support for printer's paper tray (version 4.5)

There is a new element TRAY in the Header structure of the XML payload to specify the name of the target printer's paper tray you want to use for printing. Paper trays are usually available with office printers that print on sheets of paper. They are not common with label printers that use labels on a roll.

Make sure to use the paper tray name exactly as displayed in the printer driver. When no tray name is specified, a default one from the printer driver will be used.

1.14. Support for changing label settings at print-time (version 4.5)

There is a new element LABELSETTINGS in the Header structure of the XML payload to specify the alternate label properties to use with the current print. You can modify the label settings themselves, such as label dimensions, label orientation, number of labels per row/column, offsets, and margins. You have to provide the data for the element in the standard XML notation. For more information on the structure, please see online help at: <https://help.nicelabel.com/hc/en-001/articles/360020970697-Printer#set-print-parameter-1-4>.

You can use the new option to modify label settings just before the print time, reusing the same label template for many different printouts. Using this technology, you will minimize the number of label templates you have to maintain in your printing landscape. Coupled with the support for layers and conditional objects you will minimize the need for keeping a vast number of label variants.



TIP

You can use a single label template and print it to label printers (such as Zebra, SATO, Intermec, and many others) as well as to office printers (such as HP, Epson, and Canon). For label printers, the template is printed on a roll of labels. For office printers, the same label is printed on an A4/Letter media in rows and columns and even with a different orientation.

1.15. The spool requests in SAP are linked with your output devices (version 4.5)

When you request a print stream (e.g., ZLP if you use Zebra printers), the ABAP Package receives the print stream from Software Automation and stores it in the SAP Spooler as binary data. Each such spool request item in the SAP Spooler uses the output device **NiceLabel Raw Printer**. This output device has been transported together with the ABAP Package, so it exists on each SAP system with the ABAP Package installed and is safe to use.

With this release, you can provide the name of your own output device that wish to use in each spool request. You can quickly transport the print stream to the correct printer without changing the output device at the print time.

Output controller: Info. about spool request 534479 in system ECC

Number: 0000534479
Title: NiceLabel Request
Name: LIST NLBN /NICELAB/SAP
User name: SASO
System: ECC Client: 800
Gen. Date: 25.09.2023 Time: 09:10
Mod. Date: 25.09.2023 Time: 09:10
Number of pages: 1

Spool attributes Output attributes TemSe attributes

Output Device: NiceLabel Raw Printer
Format: X_PAPER ABAP/4 list: Default list formatting
Doc. Category: LIST
Recipient:
Department:

You can use your own Output Device

2. Demo transaction

2.1. Filtering printer lists by selected Automation server

The ABAP Package remembers server/printer name combinations to know which printers are available from which Automation server.

The drop-down printer list only displays printers available on the Automation server you select. The destination server depends on your connection type (SOA/RFC) and the logical port/RCF destination name. When you select a different destination, the “Printer Name” drop-down list filters the list to display printers from the destination you select.

2.2. EXPORT_DATA field available in the UI

The demo transaction includes an option to control the new **EXPORT_DATA** flag in header elements. You can enable this field to test executions of actions you define for EXPORT_DATA placeholders in the Automation configuration.



NOTE

When this field is **True**, Automation runs all the custom actions you define in the **EXPORT_DATA** placeholder in the Automation configuration.

2.3. Clearing previews when selecting new label templates

Selecting a new label template in the “Label Name” drop-down now clears the preview pane. Clearing previews prevents misunderstandings when you select new label templates and the preview still shows the old labels.

3. Automation configuration

3.1. Support for Software Document API

Software introduces a new **Document API** for accessing and controlling the Document Management System in Software Cloud.

The Automation configuration now uses the **Document API** to provide a catalog of available label templates. Previously, Automation relied on the external command-line utility to create a catalog. the Document API streamlines label catalog generation, which is now generated as needed and provided in a synchronous response. Previously, the label catalog was generated in a separate asynchronous trigger and could take several minutes to create.



NOTE

Support for **Document API** is currently only available for Software Cloud. The Automation configuration for the on-premise Software Label Management System still uses the slower command-line utility.

3.2. New placeholder support for inbound and outbound messages (updated in 4.5)

The Automation configuration now supports new custom elements in:

- **Inbound messages from SAP** to control the execution of your custom actions in Automation.
- **Feedback messages to SAP** to provide your custom data back to SAP.

•



NOTE

From version 4.3, the ABAP Package and associated Automation configuration have support for ten custom inbound and outbound fields.

3.3. Support for label copies

This release updates the **PRINT**, **PREVIEW**, and **PRINT_JOB** methods with a setting for defining the number of identical label copies you need for each label. You use the value from the **QUANTITY_IDENTICALCOPIES** field in XML headers in the print and preview actions.

3.4. Inbound XML message samples

We document inbound/outbound XML message structures in the ABAP Package Implementation Guide and provide sample XML payloads in the ABAP Package ZIP file. Samples help system integrators better understand message transport mechanisms and manually test Automation configurations.

3.5. Sending label preview to a list of email recipients (version 4.5)

The XML payload from the ABAP Package has been extended with a new element **EMAIL_RECIPIENTS** in the Header element.

The field can provide a comma-separated list of email addresses that will each receive an email message with a PDF of a label preview attached to it. The details around the email server settings, message content, subject line, and From address can be configured inside the Automation configuration or dynamically provided together with the rest of the label data in the XML payload.

4. Bugfixes

4.1. Demo transaction can't access and display Software logo

PROBLEM

This error occurs directly after calling the transaction. You have no option to run the sample and immediately get an error message.

CAUSE

You're not authorized to read from the repository that contains the Software logo. The demo transaction does not handle the exception and crashes.

SOLUTION

We updated the demo transaction to handle the error and display an appropriate error message.

4.2. Applying the value of `_PATH_DATAFILES` parameter (Version 4.2)

One of the available methods in the Automation configuration is the DATAMODEL. It will generate a field catalog (NLVR file) out of the key-value pairs provided in the payload. The label designer can import the field catalog into Software Designer to generate a list of available variables.

When requesting the datamodel you can provide a custom path for storing the field catalog. This path is now correctly used in the Automation.

4.3. Write to SAP spool fails from your own class (version 4.5)

When you create a new class using the superclass `/NICELABEL/CL_INTERFACE_ROOT` and want to write to SAP spool (parameter `I_WRITE_SPOOL = 'X'` in the `DO_PRINT_XML` method) from your class, you get a dump (system crash).

The `/NICELABEL/CL_INTERFACE_ROOT` superclass calls the `CONV_REQUEST_2_CONTENT` method of the `/NICELAB/CL_IF_SPOOL` class. In this method, the variable that causes everything to crash refers to the demo class. As it is hardcoded in the method, you can't write to SAP spool from your own class.

The issue has been fixed.

4.4. The error message is raised when no Data element is provided for the PRINT, PREVIEW, or PRINT_JOB methods (version 4.5)

The internal Automation configuration for the ABAP Package requires that your XML payload from the ABAP Package contains at least one 'Data' element in order to run the methods PRINT, PREVIEW, or PRINT_JOB. When you execute these methods without any 'Data' element, the Software Automation did not execute the required actions but also provided no feedback.

With this release, Software Automation will provide an error response, when you provide no 'Data' element with the above-mentioned methods. When your label template does not require any dynamic value at print time as you have all information hard-coded in the label template, you can leave the 'Data' element empty, it just must be there.



NOTE

You must provide the '**Data**' element in the `<Data></Data>` syntax, not in the `<Data />` syntax.

The methods other than PRINT, PREVIEW, or PRINT_JOB do not require the 'Data' element in the XML payload. You can omit it.

5. Important upgrade notices

Support for multi-server landscape

ABAP Package V4 supports multiple Automation endpoints, not just one. The process configuration (business rules) table (**/NICELAB/V_IF_PR**, also accessible via tcode **/NICELAB/IF_PROC**) is expanded to support the endpoint configuration. Each item in the table also defined the field *OperMode*, where you define the type of the endpoint, and the field *Endpoint ID*, where you define the name of the endpoint. For example, the *OperMode* **HTTP Request** and *Endpoint ID* **EMEA** would send a request to Automation using outbound RFC call to an RFC destination EMEA (as defined in tcode SM59).

When you upgrade to ABAP Package V4 your existing process configuration table will be expanded with new fields. The values for the new fields will be taken from your existing interface configuration (tcode **/NICELAB/IF_CTRL**), so there should be no upgrade issues.

The print programs provided with the ABAP Package have also been updated to support the new process configuration table.

Label catalog generation

When you have a Software Cloud product, Automation uses Document API to create a label catalog, not the command-line utility as in previous versions. The Document API provides a faster and synchronous response. To use the Document API, you have to provide the subscription key in your payload. Automation will use it for authentication when accessing the Document API. For more information, see the chapter **Requesting label catalog for Software Cloud** in ABAP Package Implementation Guide.

The content of Revisions fields in the label catalog

In V4, Automation will return the version number of the last revision of the label template. The field will contain a single value, no longer all available versions. The returned version number depends on the access role to which the user running the Automation server belongs.