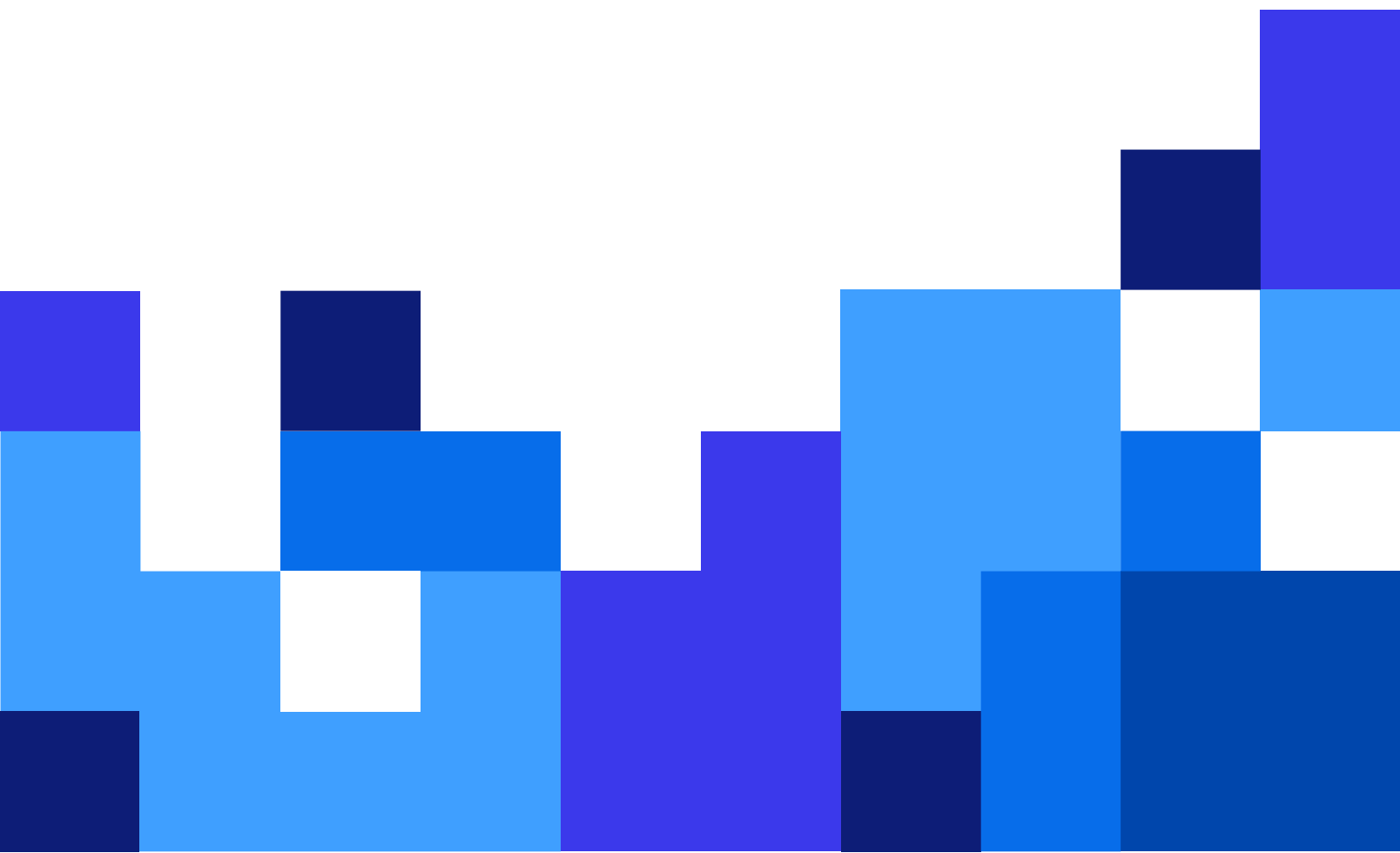


# Loftware ABAP Package

## Frequently asked questions

©Loftware 2024

Rev-10



# Contents

SAP SYSTEMS.....	3
ABAP PACKAGE BASICS .....	6
INSTALLING AND UPGRADING.....	10
INTEGRATING ABAP PACKAGE.....	11
USING THE ABAP PACKAGE.....	14
PRINT HISTORY.....	18

# SAP Systems

## What are the differences between SAP Business Suite, ERP, ECC, and EWM?

**The SAP Business Suite** software is an integrated suite of modular applications designed to perform essential end-to-end business processes across all industries. The SAP Business Suite supports a broad range of processes for finance, human resources, manufacturing, procurement, product development, marketing, sales, service, supply chain management, and IT management.

**ERP** stands for “Enterprise Resource Planning”. Many companies provide ERP software.

**ECC** is an ERP software provided by SAP SE Company. ECC stands for “ERP Central Component”. The entire SAP ERP contains other components as well (next to the ECC).

*NOTE: In 2004, SAP renamed its core ERP product to SAP ERP Central Component (ECC.) SAP ERP and ECC are now used interchangeably.*

SAP Business Suite is a collection of many SAP applications. ERP is a collection of many “installable units”:

- SAP Business Suite Applications = SAP ERP + SAP CRM + SAP SRM + SAP SCM + SAP PLM
- SAP ERP = SAP ECC + SAP XSS + SAP BW + SAP EP + SAP PI + SAP NetWeaver MDM + SAP NetWeaver AS

## What are the differences between SAP HANA and SAP S/4HANA?

**SAP HANA** is an in-memory database designed to accelerate the SAP applications and the business processes they support. HANA design innovations such as in-memory operation and a simplified data structure improve performance, leading to higher productivity, real-time visibility, and better use of analytics. For example, companies can react to market conditions based on up-to-the-minute data, rather than relying on the last quarterly reports.

While the previous versions of SAP ran on Microsoft, Oracle, IBM DB2, and similar databases, the new-generation SAP runs on SAP HANA. This also raises the hardware requirements for the SAP system.

**SAP S/4HANA** is an entirely rebuilt ERP solution (SAP ECC) that takes the advantage of HANA’s performance and innovative data models. S/4HANA provides the award-winning UX (SAP Fiori) along with a host of application and use case-specific benefits. S/4HANA can also run the old SAP GUI.

S/4HANA is available as a multi-tenant cloud offering or on-premise. The ABAP Package can work with the on-premise variant, but not with the cloud variant (because the user does not have permission to transport the ABAP Package to cloud S/4HANA).

SAP HANA is a platform, while SAP S/4HANA is a product.

A good article covering the roadmap from SAP ERP 6 to SAP S/4HANA is [here](#).

## What are SAPscript, SmartForms, and Interactive Forms?

These are all different document-output technologies that SAP provides to design and print documents (and labels). Chronologically, SAPscript came first (1990), followed by SmartForms (2000) and Interactive Forms (2006).

They are mainly focused on form/document design and are “misused” for the label design. They support office printers (e. g., to print A4 documents), but have poor support for label printers and printing labels on a roll.

- **SAPscript** is no graphical label designer. You have to “design” your label by manually typing in the “printer commands”. Essentially, you have to program the label in the printer programming language. For Zebra printers, you must use the ZPL. To print a TrueType font on a label, you must first download the font to the printer. All label design changes require label reprogramming.
- **SmartForms** introduced a document designer, but there is no WYSIWYG effect. SmartForms is not dedicated for barcodes or labels. There is limited support for thermal printers, resulting in “GDI printing”. All label objects print as graphics. You still cannot use internal printer functionality, such as printer fonts, counters, RFID, etc.
- **Interactive Forms (or Adobe Forms)**. The designer is Adobe LiveCycle Designer. As with other products from Adobe (such as Photoshop or InDesign), this one is not a tool for business users, but IT professionals. It is a very capable document designer but is an “overkill” for label design. It is also not envisioned as a label designer and lacks proper support for thermal printers. It does have some optimization techniques (e. g., for using the printer fonts and barcodes), but it only supports some major printer brands, and offers limited support for barcodes and data-encoding standards.

All these technologies offer very limited support for internal printer functionalities that we take for granted with Software software. The label design process is awkward and difficult. It takes skillful SAP professionals to use and configure these technologies. Label designing is an expensive process that takes a lot of time to complete (e. g., thousands of Euros and weeks for simple label changes, such as object repositioning or adding new objects to a label).

## What are RDI, XSF, and XFP?

All these acronyms are raw data exports that can be configured in SAP document-output technologies. They allow you to take the existing form/document and strip it of all the formatting and non-necessary layout information (fonts, colors, positions), and then provide the raw data into an external output management system that parse the received data and print it out on its layout (labels).

Software Automation can take the role of such an output management system. Software Automation can receive the RDI, XSF, or XFP data, analyze it with its filter and print the data on Software NLBL label templates. NiceLabel LMS Enterprise is [certified by SAP for compliance with the BC-RDI](#) scenario. We can also process XSF and XFP data (but haven’t certified the product yet).

- RDI. Raw Data Interface for SAPscript forms.
- XSF. XML for Smart Forms. XDF is another XML-based output that Smart Forms can produce.
- XFP. XML for PDF-based forms.

Processing the RDI, XFS, and XFP data is an alternative integration approach to the ABAP Package. It requires no coding on the SAP side (just some minor configuration), but only allows label printing. There is no support for label preview or other data responses from Software Automation.

## What is the /NICELAB/ namespace?

/NICELAB/ is the unique name of a namespace that Software registered at SAP. All objects that are included in the ABAP Package are defined within the /NICELAB/ namespace.

The namespace allows us to deliver ABAP Package to any of our customers and be assured that our custom development remains protected from unauthorized customer modifications, and SAP patch applications and upgrade procedures. The ABAP Package is transported into its own “container” separated from the rest of the SAP system.

The namespace is used as a prefix for the naming of all objects in the ABAP Package. For example, the demo application has a name /NICELAB/IF\_DEMO, and the main class has a name /NICELAB/CL\_INTERFACE\_ROOT.

## What are BAdI, BAPI, “user exits” and “customer exits”?

All these abbreviations are programs that enable us to extend the capabilities of SAP systems to meet custom business requirements. Since specific industries often require special functions, SAP allows you to predefine these points in your software.

- **BAdI (Business Add-In)** is a custom enhancement of standard SAP systems. You can insert them into the system to accommodate user requirements that are not very common and should not be present in the standard configuration of SAP. For instance, in some industry branches, you may have a specific requirement for a business process that cannot be covered by the standard SAP functionalities. This requirement can be addressed by creating a special BADI that will implement the missing functionality.
- **BAPI (Business Application Programming Interface)** is an interface that provides access to the data and processes inside the SAP systems. First, the data and processes can be defined as API methods for SAP business object types. Second, they can be defined as interfaces that are implemented outside the SAP system but can be called from within the SAP system.
- **User exit** is a point in the SAP application program planned by the SAP Developers, where you can insert your code. Inside a user exit, it is possible to read and change almost any data (local or global) from the host SAP program. Therefore, user exits give you a lot of flexibility, but this flexibility comes at a price of the higher risk to make a critical error that would lead to an ABAP dump or inconsistency in database records.
- **Customer exit** is an ABAP function that is called by the standard SAP programs. It serves the same purpose as User exit – Customer exit enables the users to add their functionality to the standard SAP transactions. Unlike User exits, Customer exits are more restrictive in terms of what you can do with them.

## What is an “enhancement spot”?

Among the SAP software advantages are the possibilities to adapt the software to your requirements, and the possibility to keep the adaptations during the upgrade. Enhancement spots allow you to add custom functionalities to the standard SAP programs.

For example, we use the enhancement spot in an ABAP Package to print labels from the standard outbound delivery transaction (e. g., transaction code VL02N). When a user initiates printing of a standard form, the enhancement spot detects the print activity and triggers the print program. The same data that is printed on a standard form (A4 document) is captured, sent into ABAP Package, and printed to a label printer.

We can demo the enhancement spot in our live SAP ERP system.

## What are the differences between S/4HANA deployment options?

There is good reading in the following articles from SAP Blog:

- [Difference between SAP S/4HANA Any Premise, SAP S/4HANA Cloud Essentials, and SAP S/4HANA Cloud Extended](#)
- [SAP S/4HANA Cloud and On-Premise Deployment Options](#)

# ABAP Package Basics

## How to get the ABAP Package?

ABAP Package is the add-on package available with **Software Cloud Business**, **Software Cloud Compliance**, **Software Cloud Enterprise**, and **NiceLabel LMS Enterprise**.

For more information, please contact your Software partner or Software representative.

## What are the requirements for using ABAP Package?

- SAP products based on the NetWeaver stack (e. g., ECC/ERP, SCM, EWM).  
Enhancement Package 6 (EHP6)  
SAP\_BASIS and SAP\_ABA system components must be at least in release 731.  
or  
On-premise SAP S/4HANA  
or  
SAP S/4HANA Private Cloud (where you can install ABAP Package as an add-on)
- Unicode SAP system

## Does the ABAP Package run in SAP EWM and use PPF?

SAP Extended Warehouse Management (EWM) is still ABAP-based so we do support integration on the EWM system.

SAP EWM does not use the “message control” mechanism to trigger output as used in SAP ERP but uses the Post Processing Framework (PPF) and print condition records. Condition record can be based on the process type or person name or other condition/event.

PPF provides SAP applications with a uniform interface for condition-dependent generation of actions. The actions are generated if specific conditions are fulfilled for an application document, for example, a specific status is set (approval by some person) or a specific date has been reached (two weeks before the end of contact).

At this point, we do not have the built-in support for PPF inside the ABAP Package. The customer receives just a bare API and must build a little bit of interface around it in PPF. The user would have to configure their printing function modules in PPF from where the printing is triggered. It is a standard integration using standard configuration tables inside PPF.

The ABAP Package can be integrated into a user-exit or BAdI available inside the EWM PPF transaction. See the **Implementation Guide**, chapter “Configuring transaction to call ABAP Package”. The usage example is also delivered as a part of the ABAP Package with the program /NICELAB/INTERFACE\_DEMO.

More information:

- [SAP Business Process Management](#)
- [SAP Help Portal](#)

## Does ABAP Package run in SAP S/4HANA Cloud Essentials Edition (EE) (previously known as Multi-tenant Edition (MTE))?

No. S/4HANA Cloud Essentials Edition does not allow uploading custom ABAP add-ons. At this time, SAP S/4HANA Cloud MTE does not allow you to “push” data to the external system.

However, to print labels, you can use the data “pull” method, where Loftware connects to reads the data from the SAP system and merges it with the label templates. The customer must configure and enable an API that Loftware consumes (HTTP REST). SAP already exposes many APIs (<https://api.sap.com>) and customers can create their own.

## Does ABAP Package run in SAP S/4HANA Cloud Extended Edition (EX) (previously known as Single-tenant Edition (STE) or Private Cloud)?

Yes. S/4HANA Cloud Extended Edition allows uploading custom ABAP add-ons, such as ABAP Package, and has more flexibility, and offers more client control than the Essentials Edition. You must have the ability to do code modifications to call methods in ABAP Package from your applications.

## Does ABAP Package run in SAP ME?

No. SAP ME is built on J2EE architecture (Java), not in ABAP. You cannot install ABAP Package in SAP ME.

## Does ABAP Package run in SAP Business ByDesign?

No. SAP By Design is a cloud-based business solution designed for small and medium-sized businesses that do not want to install anything on their IT infrastructure. As a cloud system, it does not allow users to modify the existing system. You cannot transport ABAP Package in SAP ByDesign.

## Does the ABAP Package run in SAP Business One?

No. SAP Business One is business management software (ERP) designed for small and medium-sized enterprises but does not run on ABAP.

## Does the ABAP Package run in SAP HANA 2.0, express edition?

No, SAP HANA express edition provides just HANA database and is intended for development on HANA DB. There is no ABAP backend, so the ABAP Package cannot run in this environment.

## What is the ABAP Package?

The ABAP Package is “Loftware API for label printing in SAP”. It provides a set of methods that SAP transactions execute to print labels. Some of the methods are “print label”, “get label preview”, “check printer status”, and many others.

The purpose of the ABAP Package is to establish communication between SAP and Loftware ecosystems. SAP holds the master data for labels and knows when the label should print to which printer. Loftware includes the Document Management system, role-based access control, revision control system, QA workflow management, easy-to-use label designer, and integration system, capable of printing labels on-demand.

The lack of label printing capabilities in SAP is compensated with the ABAP Package. You do not have to establish a communication route and protocol from scratch, but re-use what the ABAP Package provides. We spent months developing the ABAP Package interface, so you do not have to.

## Does ABAP Package require any additional SAP licenses?

ABAP Package does not need any SAP licenses on its own. As an API, it executes within the session of the already logged-in SAP user. The license is required for that session, not for the ABAP Package. ABAP Package creates an outbound connection to Software Automation. There is no inbound connection into the SAP system (which might require a license).

You do need additional licenses for your Software product. Please contact Software Sales for details.

## What is the difference between the Software integration system and Software Automation?

Software integration system is the platform for implementing label printing into existing business systems. The main goal of any integration is to ensure accuracy and efficiency by interfacing with the master data. The existing business system handles the master data while the Software integration system guarantees optimal and fast printouts.

Software Automation is the name of the integration system provided in Software Cloud products. Software Automation uses “triggers” to accept data from any protocol, “filters” to extract data from any modern and legacy data formats, and “actions” to build business rules and workflows. The intuitive graphical integration builder allows building integrations without coding.

## What is the SAP label printing workflow when using ABAP Package?

1. The “print program” in SAP transactions collects the data needed for label printing (name-value pairs).
2. The “print program” sends data to the ABAP Package and includes the information about which label printing is requested, the name of the label, the name of the printer, and the required label quantity.
3. ABAP Package converts the received data to XML payload.
4. ABAP Package sends XML payload to Software Automation.
5. Software Automation processes the XML data, reads a label from DMS, creates a print job, and sends the finished label to the printer.
6. Software Automation provides feedback to the ABAP Package.
7. ABAP Package provides feedback to the “print program”.

## Is the ABAP Transport Package strictly Client dependent or does it contain Client independent parts as well?

The ABAP Package contains client independent objects (workbench objects) only. The package also delivers some maintenance view entries for Software API configuration. An example of such a table is /NICELAB/IF\_CTID that contains necessary configuration IDs for the API. As this table is defined as cross-client (no MANDT), it is not necessary to import it into all target clients.

It is enough to import the initial transport into a single target client only (in cases you have multiple clients created on an SAP system), as workbench objects are cross-client. However, when doing configuration for the Software API, the client-specific views need to be maintained on each client separately.

## Is the code behind enhancement spots for Outbound Delivery called for all deliveries?

The ABAP Package enhances two standard print programs for Delivery notes through Implicit enhancements (RLE\_DELNOTE, RVADDN01).



Printing from the enhancement spots is only executed if enhancements have been activated via the enhancement spot configuration table (transaction /NICELAB/IF\_ENHS). Furthermore, printing will only be executed if the configuration for an outbound delivery process is maintained on Output Type and SalesOrg level (transaction code /NICELAB/IF\_PROC).

## **ABAP Package contains an append to standard structure MSFCV. Can this cause future support issues?**

There is an append to standard structure MSFCV to add a single field (description of material type). The structure is used in multiple SAP standard programs.

This is a Software append structure and the field lies in the Software namespace, so it will not have an impact in the future releases as /NICELAB/ namespace is officially registered with SAP. There is no risk that SAP or a third party delivers a field with the same name.

## **Why does the ABAP Package directly update the table TSP01 (spool requests table)? Is this best practice?**

Software logs all data about the print history inside Software Document Management System. Alternatively, you can also enable logging on the SAP side. In that case, the ABAP Package will log outbound/inbound messages in SAP Spooler. You can log just status feedback, or whole responses from Software, such as label previews (e. g. PDF), or print jobs (e. g. ZPL for Zebras).

ABAP Package updates the spool requests status, but only for the spools which are generated by Software API. The reason for this is because the actual printing does not happen in SAP but on the Software side. To have the actual information (from Software Automation that drives the printing) the update of spool request status is needed.

This is an optional feature that is disabled by default.

# Installing and upgrading

## How to install the ABAP Package?

ABAP Package is a transport request. It follows the standard method that defines how the add-ons get installed in the SAP landscape.

To install the ABAP Package:

1. Unzip the ABAP Package file.
2. Copy the transport files to the appropriate folders in the SAP file system.
3. Run the **STMS** transaction code.
4. Initiate the transport of the package.

For details, see chapter **Importing ABAP Package transport request** in the Implementation guide.

## How to upgrade ABAP Package to a new version?

To upgrade ABAP Package, you must follow the same steps as when you imported it the first time, you just have to enable the option “Overwrite Original” inside the STMS transaction code.

For details, see chapter **Upgrading ABAP Package** in the Implementation Guide.

## We plan to upgrade S/4HANA to a newer version. How does the SAP upgrade affect the ABAP Package?

The core of the ABAP Package is a self-sufficient API that survives SAP upgrades. In most cases, you don’t have to do anything for the ABAP Package after you upgrade your SAP.

However, ABAP Package also contains objects outside of the core API, such as enhancements and demo transactions that help you either to test the functionality or to easily integrate the ABAP Package. These objects are linked to specific SAP components.

For example, ABAP Package contains the enhancement spot for Outbound Deliveries (for SmartForms and SAPscript). If the new version of SAP updates these objects that we use in the ABAP Package, the ABAP Package’s enhancement will remain in the “open” status. If you do not need this enhancement in your label print processes, you can leave it as is (open). This does not impact your production use. But administrators tend to keep statuses “closed” and want to keep the open issue list empty.

When the SAP standard objects change after the upgrade and this affects ABAP Package enhancements (such as enhancement spot for Outbound Deliveries), the ABAP Package must be “repaired”, so the ABAP Package is adjusted to the SAP upgrades and becomes compatible with the new version of SAP code.

The repair instructions are available on the SAP site:

[https://help.sap.com/doc/saphelp\\_nw74/7.4.16/en-us/bd/ddbe0bac5c11d2850e0000e8a57770/content.htm?no\\_cache=true](https://help.sap.com/doc/saphelp_nw74/7.4.16/en-us/bd/ddbe0bac5c11d2850e0000e8a57770/content.htm?no_cache=true)

# Integrating ABAP Package

## How does SAP deliver data to the Software integration system?

ABAP Package can use either of the following outbound communication channels available in SAP:

- **Web Service (SOA).** In this case, you must configure the Web Service endpoint in the transaction code SOAMANAGER.
- **RFC type G (HTTP connection to an external server).** In this case, you must configure the HTTP endpoint in the transaction code SM59.

The ABAP Package also supports communication through the PI/PO and CPI proxies.

The ABAP Package delivers the same type of XML payload through any of the communication channels.

## Can I encrypt the communication between SAP and Software?

Yes, Web Service and RFC Type G (HTTP) triggers in Software Automation support the encrypted transmission (SSL/TLS). Minimal reconfiguration of Software Automation is necessary to enable HTTPS support (just one checkbox). Additionally, you must add the SSL/TLS certificate on the computer running Software Automation and bind the computer to the port number. Instructions are available in the Software Automation help.

## Does the SAP implementor have to code the content for the XML payload?

No. ABAP Package creates the XML payload for Software Automation with the data it receives from the “print program”. The transformation with ABAP Package converts the received data (name-value pairs, method to execute, label name, printer name, label quantity, etc.) into the XML structure.

## How to use the ABAP Package from a custom transaction?

ABAP Package is an API and exposes the methods that you would execute from your ABAP code. You can integrate it everywhere in the code, where label printing is needed.

Depending on the requirement and the process, you can use one of the following approaches to send the data from the SAP application to Software Automation using the ABAP Package:

- User exit in SAP standard coding
- Business add-in (BAI) / Enhancement
- Create document output and change the print program (most popular)
- Custom ABAP program

For more information about how to do it, see:

1. Implementation guide (PDF document) that is included in the ABAP Package. See chapters “Configuring transaction to call ABAP Package” and “API Reference”.
2. See the source of the demo application that is included in the ABAP Package. Learn how the demo application consumes methods in ABAP Package and repeat the procedure.

/NICELAB/IF\_DEMO

## Can I test the ABAP Package before I commit any programming changes in SAP?

Yes. The ABAP Package includes the demo transaction to test the connectivity between ABAP Package and Software integration system.

To test the ABAP package:

1. Install Software Automation.
2. Load and activate the Software Automation configuration provided with the ABAP Package.
3. Make sure the sample labels (.NLBL files) are saved in the same folder as the configuration (.MISX file).
4. Transport the ABAP Package in SAP.
5. Configure the SOAP (t-code SOAMANAGER) or REST (RFC Type G) (t-code SM59) endpoint.
6. Run the demo transaction.  
`/n/NICELAB/IF_DEMO`
7. Test the label preview functionality. The label preview updates in real-time with the data you provide for the field values.

For more information, see chapter “Using the demo transaction” in the Implementation Guide.

## Do I need to know the variable names from the label?

No. You do not have to hand-pick the variables in SAP that exist on the label today and send just those. The best-practice approach is to extract all name-value pairs from your transaction. All two hundred or three hundred that exist in the transaction – including the ones you use in the label today, and the ones you might use in the future.

Software Automation extracts all name-value pairs from the received XML data. If the label template contains a variable of the specific name, that variable receives a value. If the variable from the received data does not exist in the label template, that variable is ignored.

The data binding is done on the matching names of fields (from SAP) to variables (in the label template).

If you export the entire data model (Field Catalogue) today, you are prepared for future needs. When a future label change request is received, you only update the label template to include the new dynamic object.

- No need to do anything in SAP.
- No need to do anything in Software Automation.
- Just update the label design.

There is no measurable processing penalty because of the slightly larger data size. Software Automation is a new-generation integration tool capable of processing huge amounts of data in real-time. The extra fields in the XML data do not slow it down.

## Can I read the names of variables defined in the label template?

Yes. You can execute the method “get label variables” that returns a list of variables that exist in the label template. At the same time, you get a detailed report on the variables’ properties.

## How does the variable mapping work in Software Automation?

Software Automation extracts the name-value pairs from the received XML payload and sends them to the label. If the variable from the name-value pair is defined in the label, it is automatically populated with a value. If the variable does not exist in the label, the field's value is silently ignored.

The names of variables defined in the label template must match the names of fields in the SAP transaction for the auto-mapping feature in the Software Automation configuration to work.

For example, the label printed from outbound delivery transactions must contain variables, such as VBDLP-ARKTX, VBDPL-MATNR, and VBDPL-MAINS, because those are the field names used in SAP.

## Is there an easy method to create variables on the labels?

Yes, there is. There is a configurable option in the ABAP Package to create a list of Software variables out of SAP field names. You can execute a "data model" method. In this case, Software Automation creates a Field Catalogue based on the received data and saves the data as an NLVR file. This is the Software-native XML file for defining the variables in the label template. You can import the Field Catalogue from the NLVR file into the label template.

This is a one-time job per SAP source of data. Once you have the Field Catalogue defined for one label, you can import this data model to other labels.

Software ABAP Package provides two sample labels (for "outbound delivery" and "production order" transactions) that already contain variables with proper names.

## Can I define default values for parameters for the ABAP Package?

Yes. Configuration table for transaction defaults is a part of the ABAP Package. Use this table to specify the default values for basic parameters. When you specify a parameter in the configuration table, you can omit the parameter when consuming the API methods. ABAP Package always reads the default values, but you can override these default values with each API call.

## Does ABAP Package support bidirectional messaging and status reports?

Yes. ABAP Package communicates with Software integration system using Web Service or RFC Type G (HTTP) communication. Both communication types are bidirectional by default. Software Automation returns real-time statuses to SAP, so you know exactly what happened during the processing of your data:

- Print job status
- Printer status
- Processing errors

Some of the methods available in the ABAP Package do not even print labels, such as "get label preview", "get a print job", "get a list of available printers", "get label catalog", or "get a list of variables from this label". In that case, you receive the requested data.

# Using the ABAP Package

## What is a “print job”?

A print job contains printer programming language for the target printer, e. g. if you use Zebra label printers, the print job contains “ZPL” (Zebra Programming Language). Software merges master data from SAP with the approved label template from DMS and creates a print job.

Software software can use any available Windows printer driver to create a print job, but it works best with Software own “[label printer drivers](#)”. After the print job is created, the Windows print system transports the print job to the target printer.

## Does the ABAP Package work out-of-the-box?

That depends on the transaction where you want to print the labels from.

ABAP Package works out-of-the-box for the standard SAP transactions, which we tightly integrated with the package. After you transport the ABAP Package into your SAP landscape, you can use the package immediately with the following standard transactions:

- Outbound delivery (VL02N)
- Production order (CO02)

The print program and/or enhancement spots are available with the ABAP Package, so you do not have to code them by yourself. However, you do have to enable the support. For details, see chapters “Example for outbound deliveries (VL02n)” and “Example for production order (CO02)” in the Implementation guide.

For any other standard or custom transaction in SAP, change the existing print programs to send the data to the ABAP Package.

## How long does it take to implement the ABAP Package?

Implementing the ABAP Package represents the programming approach of integration. You have to update your “print programs” to communicate with the ABAP Package and to execute the API methods. Completing this task takes some time.

NOTE: ABAP Package includes print programs for transactions that frequently require label printing (e. g., “outbound delivery”, “production order”). For these, you do not have to program anything in ABAP.

You have to:

1. Identify the print process. Who is the user? How do the users print labels? Is there any automation based on the events? Is this manually triggered printing?
2. Collect the data. Once the print event occurs, your “print program” must gather all data for the label. Does the current transaction provide all data for you, or do you have to gather the data from some other resource/classification?
3. Sending data to the ABAP Package. This is the easy/straightforward phase.

The ABAP Package simplifies communication with the external print server (Software Automation), as you do not have to worry about the data structure and the connection method. You just have to collect the data for printing and deliver the data to the ABAP Package when printing the labels.

This will take hours, perhaps days to complete, but certainly not weeks or months.

## Does the ABAP Package transmit a predefined list of tables/fields for a specific transaction code? Does it enable plug-and-play?

When we connect a T-code to labeling we have an enhancement spot and modified print program. We would pass all of the fields to the label template. When you design a label you would see all of the SAP fields from that transaction. You might not want to use them all, but you see them and you can easily add new if necessary.

Each T-code has specific structures behind it. We dynamically scan the structures, whether it is a print structure or internal structure where transactional data is stored and we dynamically extract all the attributes (fields and values). The data stream going from SAP into the Software integration system is presented with key-value pairs.

## Does SAP print the labels when using the ABAP Package?

No. When using the ABAP Package, SAP does not print labels. ABAP Package omits the SAP Spooler and communicates with the Software integration system directly. Software Automation creates a print job and sends the print job to the printer. The printer drivers for your print infrastructure must be installed on the computer, where Software Automation is installed. SAP receives feedback about the print status.

Optionally, you can configure ABAP Package to save your print jobs in SAP Spooler (SP01), so you can send them to the printer from SAP.

## What does a “print program” do?

A print program starts when the SAP user initiates a print request in the transaction. The print program executes in the background and has no user interface. Print programs are provided with SAP for the standard transactions that frequently require label printing (e. g., “outbound delivery”, “production order”).

The standard print program does the following:

1. Collects the data needed for printing. The data might be extracted from various definition sources in SAP.
2. Sends the data to SAP Spooler for processing and printing.

You can reuse the standard print program to use the ABAP Package. You must update the print program sections, using which the program communicates with the SAP Spooler, and redirect the data into the ABAP Package. You would reuse the existing print program to save time and not write it from scratch.

We have a sample of the modified print program **RVADDN01** (Print of a delivery note by SAPscript).

## Can SAP send a print job to a printer instead of Software Automation?

Yes. Instead of executing the “print” method in ABAP Package, you must execute the “print job” method. Software Automation creates a print job and returns the print job to SAP. ABAP Package stores the job in SAP Spooler, from where SAP can transmit it to the printer in “passthrough” mode.

## Can Software print labels use a locally-connected USB printer?

Software Automation can print labels using printers that are installed or registered on the computer, where Software Automation is installed.

- If you share your local USB printer in the network, Software Automation can print to it.
- If you do not share the local USB printer, Software Automation cannot see the printer and cannot use it for printing. However, Software Automation can create a print job and store the job in SAP Spooler.

Then you can transmit that print job from SAP Spooler to your local USB printer via SAP GUI (passthrough mode).

For more information, see chapter “Printing binary print jobs from SAP” in the Implementation guide.

## Can Software Automation combine master data from SAP with other data sources (data enrichment)?

Yes. While the best-practice approach is to consolidate the data inside SAP and to provide all data to the ABAP Package for a single message, you can configure Software Automation to fetch the data from additional data sources and merge it with the data received from SAP (i. e. “data enrichment”).

Software Automation can get data from various 3<sup>rd</sup> party sources, such as SQL data sources, can execute Web Service/HTTP REST methods, and can communicate with TCP/IP and serial port devices.

You can update the Software Automation configuration yourself, or communicate the request for service through Software partners or Professional Services group.

## Querying multiple data sources to obtain values for a label print job most surely impacts performance. How to manage that?

Yes, it can be more time-consuming, when data for a label has to be collected online from multiple sources (external systems). This is a process design question, as customers can have different approaches. It’s up to them to implement efficient data reading. For example, some customers use shadow tables to buffer the external data, which can speed up the reading process.

## Can I see the label preview in SAP?

Yes. One of the available API methods returns graphical label previews. ABAP Package uses real master data and approved label templates and sends them to the Software integration system. The label preview becomes available in the requested graphic format (e. g., PDF, PNG, JPEG) and returns to SAP.

Typically, you would use the label preview for:

- Print previews/print simulation
- Label approvals and QA workflows inside SAP
- For electronic archiving of finished labels

## Can I use the ABAP Package to export data from SAP?

While the out-of-the-box configuration of the Software integration system does not store the data from SAP, but only uses the data for real-time printing or generation of label previews/print jobs, you can easily adapt the Automation configuration to store the data instead of printing it.

The required configuration changes are minimal. For example, you can use various built-in actions to store the data on your SQL server or in a flat CSV file or to send the data to Web Service/HTTP REST destinations. The change requires no coding, just the configuration of an off-the-shelf product.

## Can I use the ABAP Package to create label variants?

Yes. You can execute the “VARIANTS” method to embed the production data in label templates and save them as “label variants”.

Label variants are label templates ready for production. They originate from the same approved label template and have the production data embedded. The only unlocked dynamic objects on the label are the



ones that are applicable/known just before the production starts, such as “LOT number” and “production date”. Label variants are still NLBL label files but are locked for editing.

For more information, see chapter “Support for label variants” in the Implementation Guide.

## How do we monitor the integration from ECC or EWM to Loftware to see any exceptions?

All requests to Loftware and responses from Loftware can be logged in the SAP spooler (T-code SP01).

The processing log is also kept on the Loftware side. You can see all details in the Automation Manager and Control Center.

## Can we empower business users to make label changes?

Yes. Loftware Designer is a Windows desktop application for label template designing. Loftware Designer deliberately uses an interface with ribbons like Microsoft Office applications. Users coming from Microsoft Word or Microsoft PowerPoint immediately find their way around Loftware Designer.

The label design process with Loftware involves graphically designing the label, not coding the label in a printer programming language. Business users can start designing labels without any steep learning curve. One of the goals of the ABAP Package is to move SAP IT teams out of the label design process if that is what you wish (not all companies want to pass label design control out of IT hands).

# Print history

## Is print history kept anywhere?

Yes. Software Document Management System (DMS) makes all print events transparent. Each time the label print action request comes from SAP, Software Automation stores the entire log of print events. DMS stores the following events:

- Label name and version.
- Name of the user that initiated printing.
- Timestamp of a print event.
- Printer name used for printing.
- Label quantity.
- Value of each dynamic object on the label.

Software DMS also includes the Analytics part, where you have an insight into printing trends in your company, such as “which labels are printed most”, “where do most error occur”, “which printer is busiest”, “which computers print labels”. The print history is not kept for printing initiated just from ABAP Package, but for all Software clients in your landscape.

## Can you save print history in SAP?

Yes. You can enable the option to save print events in SAP Spooler (transaction code SP01). Add the “WRITE\_SPOOL” entry in the ABAP Package configuration table /NICELAB/IF\_CTRL, and set its value to “True”.

Each print event is then logged in SAP Spooler. The event can also store the label preview or print job if requested.

With active default settings, print history is kept just in Software DMS.

