Oracle Warehouse Management Cloud

Document and Label Designer Guide

Release 26D

Oracle Warehouse Management Cloud
Document and Label Designer Guide

Release 26D

G58496-01

Copyright © 2026, Oracle and/or its affiliates.

Author: Oracle WMS Cloud Product Team

Contents

Get Help i

2 Label and Document Designer 3

Label and Document Designer 3
Supported Documents and Labels 3
Action Bar 4
Creating and Editing Boxes 5
Editing Label Lines 7
Calculating Pixels for Precise Label Design 10
Creating Labels and Documents 10
Column Calculation for Text Elements 12

3 Company Report Type Document Configuration 17

Company Report Type Document Configuration 17
Pre-established Templates 18
Calling Label Designer Templates 20
Calling Web Reports Templates 21

4 Price Labels 23

Price Labels 23
Editing Document and Label Pages 24

5 Creating Barcodes 27

Creating Barcodes 27
Create a Label or Document with Dynamic Repeating Barcodes 33
Detailed Barcode Editing 33
Print Preview 34

6 Multi-Field Barcode Scanning 35

Multi-Field Barcode Scanning 35

Multi-field Barcode Setup 35
Working with Multi-Field Barcode Classes (Application Identifiers) 36
Support for FNC Characters for GS1 Barcodes 42
Image Based Barcode Support 45

7 Document and Label Designer Expressions 47
Document and Label Designer Expressions 47
Expressions 47
Python strftime Directives 49
Using Functions within Expressions 50
Adjusting Timestamp using Expressions 51
Testing Your Expressions 52

8 Mappings 55
Mappings 55
Mapping Details 55

9 Standard Functions 57
Standard Functions 57
Function Details 57

10 Text 67
Text 67
Edit Text Definitions 68

11 Uploading and Editing Images 71
Uploading and Editing Images 71
Working with File Formats 72

12 Printer and Page Setting Guidelines 73
Printer and Page Setting Guidelines 73
Labels in Additional Languages 73
Multi-Page Printing 73
Print Multiple Serial Numbers 76

13 **Label Template Setup** **77**

Label Template Setup 77

Base Label Templates 77

Label Template View 78

Get Help

There are a number of ways to learn more about your product and interact with Oracle and other users.

Get Help in the Applications

Access the online help from the user drop-down menu in the Warehouse Management application.

Get Training

Increase your knowledge of Oracle Cloud by taking courses at *Oracle University*.

Join Our Community

Use *Cloud Customer Connect* to get information from industry experts at Oracle and in the partner community. You can join forums to connect with other customers, post questions, and watch events.

Share Your Feedback

We welcome your feedback about Oracle Warehouse Managment. If you need clarification, or find an error, you can direct your questions via a service request to *My Oracle Support.*

2 Label and Document Designer

Label and Document Designer

The Oracle® Fusion Cloud Warehouse Management Label Designer allows you to create custom labels and documents in a simplified interface.

When you are creating a new label or document, you need to do this in QA, UAT, or PP before making the same changes into Production.

The label designer data grid can now be setup to use metric measurements in addition to imperial measurements, which will be helpful in countries using metric.

Supported Documents and Labels

The following Labels and Documents are supported in Label Designer:

Supported Labels:

- Blind Container
- IB Container
- Item
- Location
- OB Container
- Order
- Task
- Shipping
- IB Shipment Label
- Serial Number
- Webservice Addon Label
- OBLPN Contents
- Work Order
- Price Labels

Supported Documents:

- Order Packing PDF Report
- Packing report for Outbound LPN

Disabling Buttons While Loading

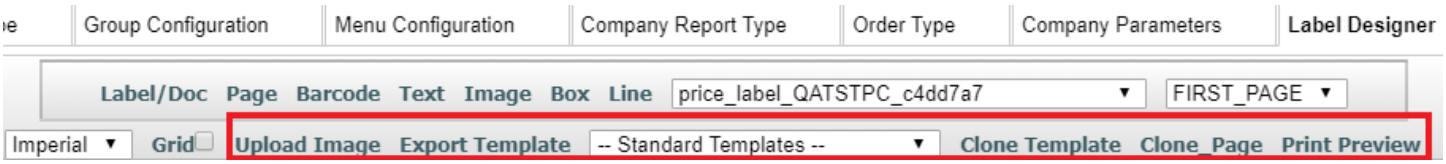
The buttons at the top making a new page, document, text, barcode, box, or line, etc. Are grayed out and disabled while it is loading.

Uploading an Exported Template

Select Standard Templates drop down and then select Upload Template. Now browse for your template on your hard drive it should be a json file. Then select the upload button. Your new template will appear in your list of current templates.

Action Bar

The Action Bar contains all of the menu functions you will need to create, edit, and configure settings for your labels and documents.



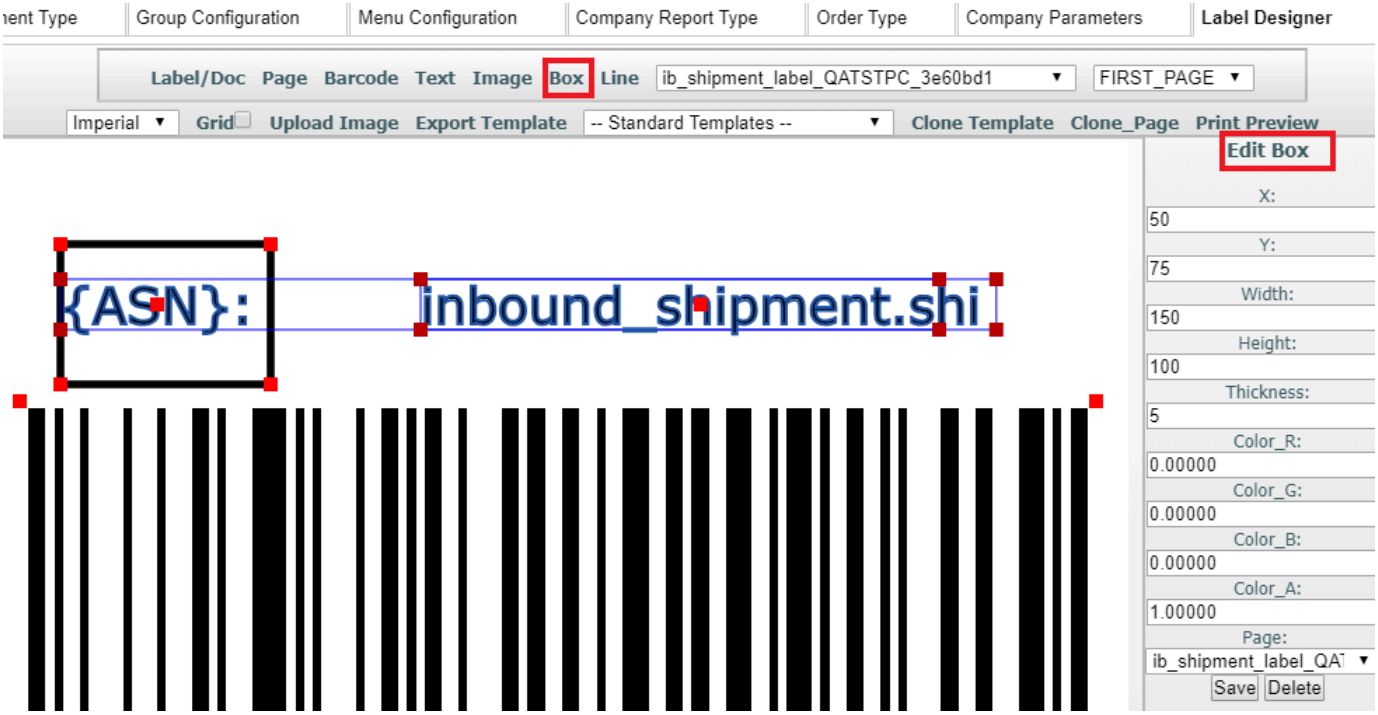
The following are descriptions of all of the action bar functions:

Action Bar Item	Function
Grid	Turns on the Grid for Imperial Units for easier alignment and placement of elements on the template.
Upload Image	Allows you to upload an image to be assigned to an image placement element.
Export Template	Allows you to export a template into a json format that can be stored and backed up. Plus this allows for a template to be developed in one environment then be transferred to another environment.
Upload Template	Allows you to import a template that you have previously exported. This will generate a new template in the template drop down. You will find this in the Standard Templates drop down.
Standard Templates	Allows you create entire standard templates for different types of labels to help save you time and headaches. You can also learn from tricks by looking at these templates.
Clone Template	Allows you to make a copy of your current template so that you can modify it for some other purpose.
Clone Page	Allows you to make a copy of your current page.
Print Preview	Creates a pdf of the mapping of your template to give you a better idea of how the template is going to be printed.

Creating and Editing Boxes

To create a new box, complete the following steps:

1. From the Label Designer tab, click **Box**.
2. To edit a box, from the Label Designer tab, click **Box**, then **Edit Box** at the top of the edit panel.

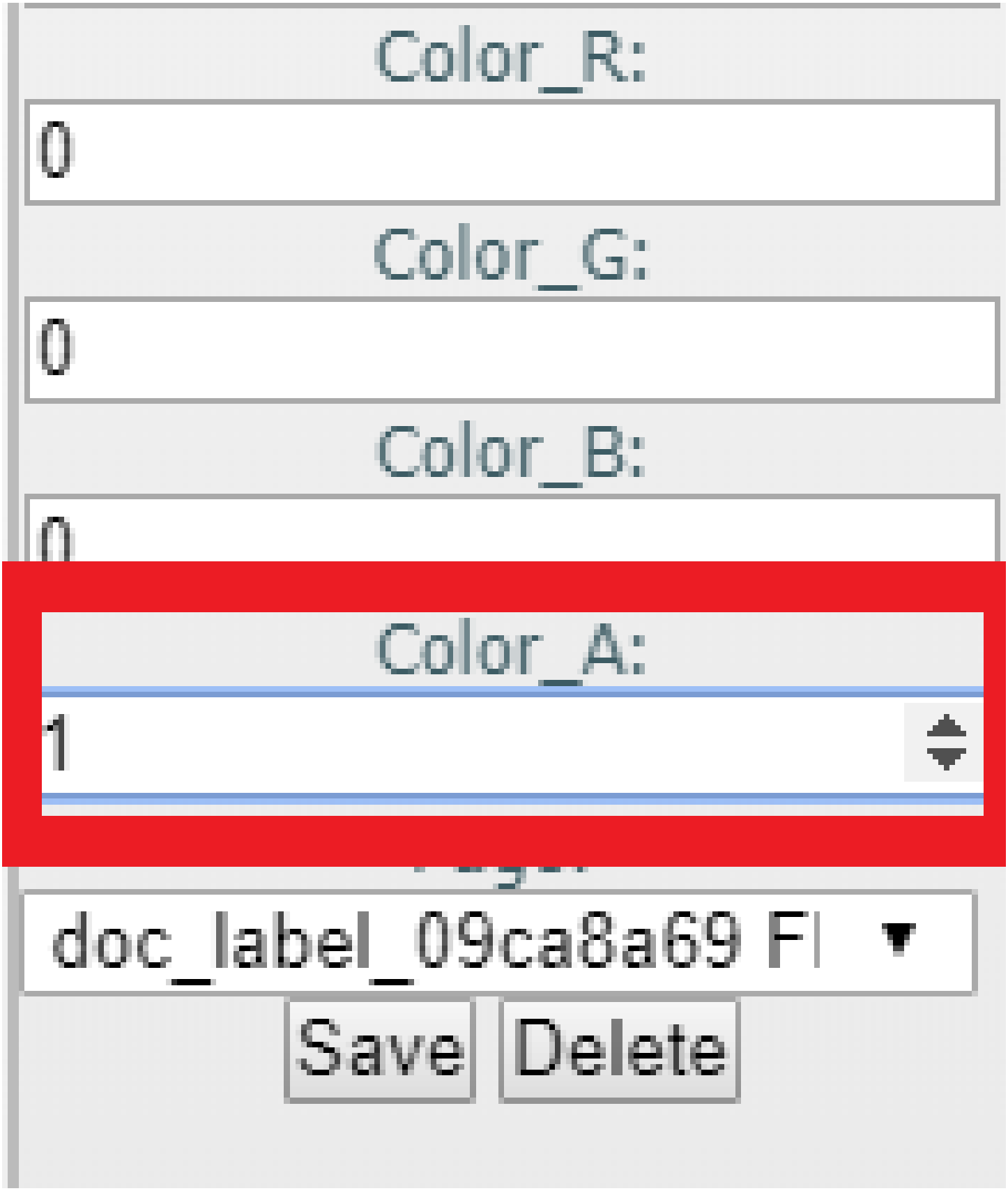


3. Change the value you want to change and then click **Save**.

To move a box, click on it and then drag the box to where you want it to be positioned. The box will automatically save its position when you are done dragging.

You can also enter X and Y coordinates to move a box. X and Y starting points are at the top left corner of the page. Width and height adjustments begin at the far right corner. You can adjust your box size by dragging and pulling or entering the number you want for positioning it. After dragging, click outside of the box to set changes and after entering values, tab to move to the next field.

If you want to make any color specifications, add a 1 in the corresponding field to indicate Red, Green, or Blue. To save your changes, click **Save**.


Color_R:
0
Color_G:
0
Color_B:
0
Color_A:
1
doc_label_09ca8a69 F
Save
Delete

Note: Zebra printers only support black and white, so you only need to specify 1 in the Color_A: (Alpha) field.

The following are descriptions for the different box settings:

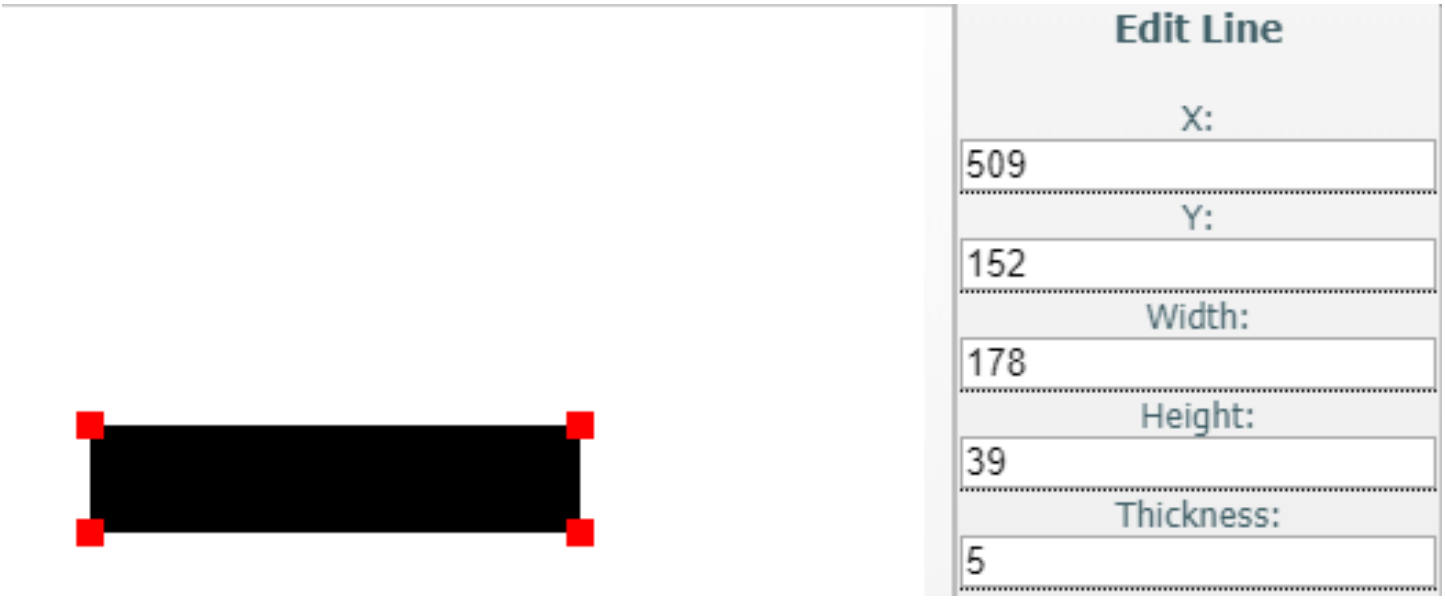
Box Specification	Description
X	Position the Box on the X-Axis based on the number of dots of ink or thermal burning on a page.
Y	Position the Box on the Y-Axis based on the number of dots of ink or thermal burning on a page.
Width	How wide your box is.
Height	How tall your box is.
Thickness	The thickness of the line that makes up the box.
Color R	How much red you want your box to be. Range is 0 - 1.0
Color G	How much green you want your box to be. Range is 0 - 1.0
Color B	How much blue you want your box to be. Range is 0 - 1.0
Color A	How much alpha/transparency you want your box to have. Range is 0 - 1.0
Page	Set what page this box will appear on.
Disabling Buttons While Loading	The buttons at the top making a new page, document, text, barcode, box, or line, etc. Are grayed out and disabled while it is loading.

Editing Label Lines

When you click on a line, the **Edit Line** option appears in the right pane, which will allow you to set your line specifications.

To make changes, change the value you want to change and then click **Save**.

If you want to create white text against a black background, you can use the Line feature to do this. You can create a line, then enlarge the meet the size requirements for your label.



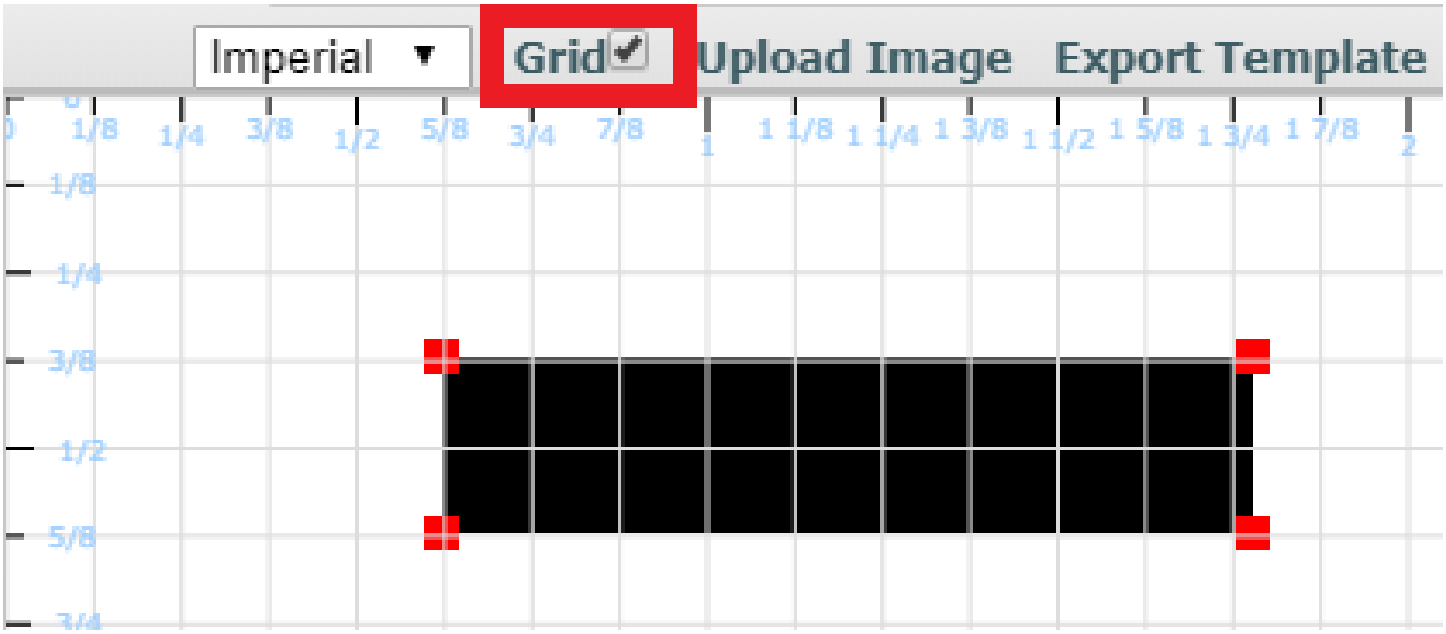
Creating a Line

Click the Line at the top of screen in order to create a new Line.

Move Line

You can move a line by clicking on it and then dragging the line to where you want it to be positioned. The line will automatically save its position when you are done dragging. Another way to move a line is by entering its x and y coordinates. X = 0 and Y = 0 is the top left corner of the page. The bottom right corner are X = page width and Y = page height. So by entering the number you want for positioning, it will allow you to put the line where you want it. After you type a number and then tab or click out of the edit line, the line will move to that location. In order to save your changes, click **Save.**

You can also select **Grid** to view imperial and metric measurements to assist you with sizing your line accurately:



Document Label Line Definitions

Label Line Item	Description
X	Position the Line on the X-Axis based on the number of dots of ink or thermal burning on a page.
Y	Position the Line on the Y-Axis based on the number of dots of ink or thermal burning on a page.
Width	How wide your line is.
Height	How tall your line is.
Thickness	The thickness of the line.
Color R	How much red you want your line to be. Range is 0 - 1.0
Color G	How much green you want your line to be. Range is 0 - 1.0
Color B	How much blue you want your line to be. Range is 0 - 1.0
Color A	How much alpha/transparency you want your line to have. Range is 0 - 1.0

Label Line Item	Description
Page	Set what page this line will appear on.
Disabling Buttons While Loading	The buttons at the top making a new page, document, text, barcode, box, or line, etc. Are grayed out and disabled while it is loading.

Calculating Pixels for Precise Label Design

Calculating pixels can be very useful for precise label and document design. Exact placement for labels and documents allows timely approval and will help you to avoid unnecessary costs.

To create precise design, you will want to set Dots Per Inch (DPI) specifications with numbers rather than dragging to adjust the item. In most cases, you will be converting imperial measurements to metric.

Note: the DPI metric conversion is 2.54 cm per inch.

Example One:

You have a Zebra printer with 203 DPI which is standard, (that is the most common DPI for a Zebra printer), you will divide 203 / 2.54 = 79.92125. There are 10 mm in a cm, so that means roughly 8 dots or pixels will go into a mm.

If you need to be over by 5 mm then multiple 5 * 8 = 40, you can move your element of text or barcode to 40, and be precisely 5 mm over from the edge of the paper.

Example Two:

For a DPI of 300, you will do the same kind of calculation:

300 / 2.54 = 118.11

Divide by 10 and you get roughly 12 pixels per mm.

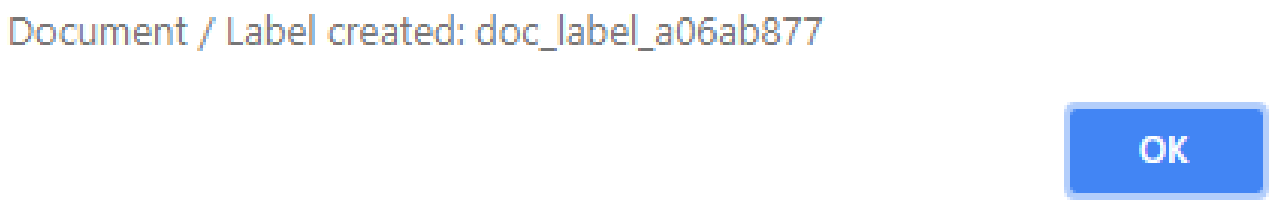
Note: Desktop printers can be set to any settings as they are not as standardized as Label printers.

Creating Labels and Documents

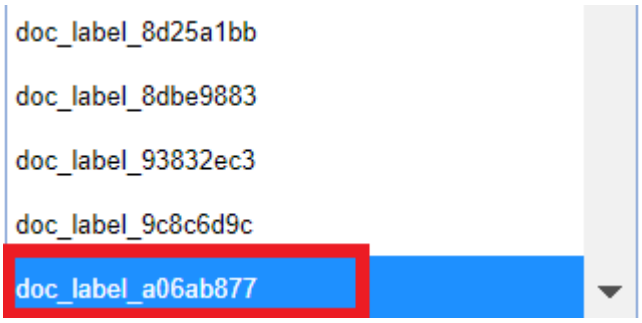
Label Designer allows you to configure settings for Labels and Documents. To configure your settings, complete the following steps:

1. To create your label from the Label Designer, click **Label/Doc**.

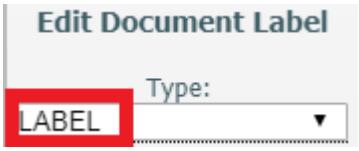
A pop-up window appears notifying you that your label has been created.



2. Select the label you just created from the **User Defined Templates** drop-down:

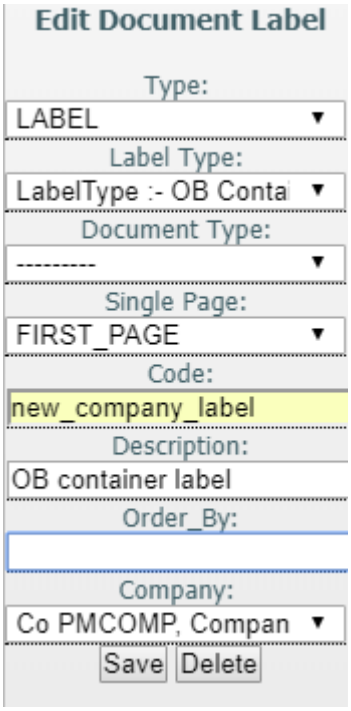


3. Select your type (label or document) from the **Type** drop-down.



> **Note:** You can't have Document Type and Label Type selected at the same time because your template will either need to be a document or label template.

4. Select your **Label Type**.



5. Enter your label name in the **Code** field.
6. Add a description for your label in the **Description** field. This is particularly useful to help you identify labels if you manage multiple labels.

   The **Order By** field allows you to order the data as it comes in from the database. There is also a company drop-down if you are managing templates for multiple companies.
7. Once you have finished adding specifications for your Label or Document, click **Save**.

Note about ZPL Zebra Labels

In some instances, underscores have caused printing issues when processing the ZPL (printer language) for Zebra Labels. To avoid data being looked up as unicode for ZPL Zebra labels inside the printer, Document and Label Designer will convert _ to the - character before doing language unicode encoding for unicode characters to print the correct characters on paper labels.

Column Calculation for Text Elements

Document Label Designer allows you to design customer specific label templates for various label types. You can sub-total a field in the details section of a label. This allows you to consolidate and display numeric fields while designing any label or document.

The Column Calculation drop-down for text elements allows you to consolidate quantities in numeric fields. An example of this would be to consolidate quantities in a field per SKU in a pallet, irrespective of the number of LPNs present on the pallet.

The following options are available in the Column Calculation drop-down:

- Count
- Sum
- Avg
- Min
- Max

When you are populating the fields from the Edit Text menu, once the **Column Name** field is populated and the data type is populated with Dynamic Repeating, the **Column Calculation** drop-down will enable a list of options:

Edit Text

Font Type:

HELVETICA

Data Type:

DYNAMIC_REPEATING

X:

-44

Y:

60

Size:

-2

Length:

Once you have created your label with a column calculation option, you will be able to see the quantities by SKU in the label.

You can design the label and fetch the total quantity of a SKU for a pallet with multiple LPNs with the same SKU (Single SKU Pallet) or Multiple SKU LPNs with different SKUs. Column Calculation is supported for all templates and all labels and documents in Document Label Designer.

Note: the Column Calculation field is numeric only. And the data type must be a dynamic repeating field. Some common business scenarios that Column Calculation could be useful for include the ability to calculate the following:

- current quantity / sub total units
- sub total cost for each item
- sub count (for example, how many containers have a particular item on the pallet)

Example

The following scenario explains how quantities would display for a pallet label based on the option selected from the Column Calculation drop-down:

Let's say you have a pallet label with 10 containers on a pallet. Each container has the same three items but each item has varying quantities. You also configure the item code to display on the pallet label along with the inventory current quantity as part of the same table name.

Note: Fields from other tables can be used for dynamic repeating elements. However, annotations can not be used by themselves and get good results. When no other fields are paired with the annotation, then all fields get selected and most likely nothing will get annotated in accordance with the requirements of the fields.

The following quantity would display based on the respective drop-down item:

Column Calculation Drop-Down Selection	Quantity Displayed in Pallet Label
Count	Quantity of 10 displays for each of the three items.
Sum	The sum of the quantities displays for each of the item's current quantities across all of the containers per item code. So you will see the sum for each of the three item codes.
Average	The average quantity displays for each of the item's current quantity across all of the containers per item code. So you will see averages for each of the three item codes.
Min	The minimum quantity displays for each of the item's current quantity across all of the containers per item code. So you will see a minimum value for each of the three item codes.
Max	The maximum quantity displays for each of the item's current quantity across all of the containers per item code. So you will see a maximum value for each of the three item codes.

Column Calculation Drop-Down Selection	Quantity Displayed in Pallet Label

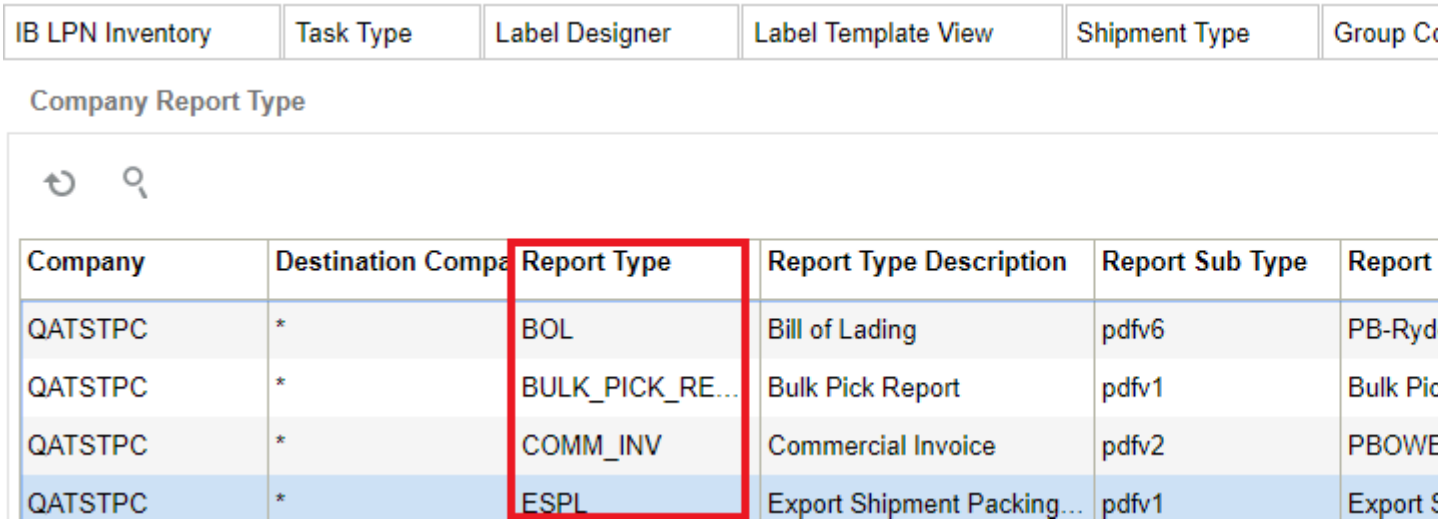
Note: If you are setting up documents or labels using multiple template pages, each field has to be associated with the same table name, or the fields won't be associated with the same fields on the different template pages. The fields that you put in dynamic repeating template pages affect the granularity of your calculations.

3 Company Report Type Document Configuration

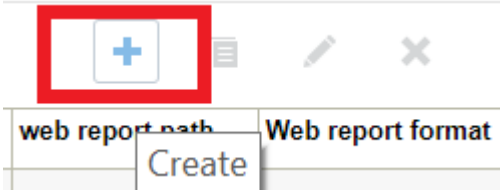
Company Report Type Document Configuration

The **Company Report Type** tab allows you to configure all available Report Types to corresponding templates. These templates can be further filtered by available fields such as Destination Company, Order Type, Ship Via, and so on. Oracle WMS currently has 13 different reports that can be generated in multiple UI screens and this configuration process allows you to call templates from Label Designer, Web Reports, and the available internal templates.

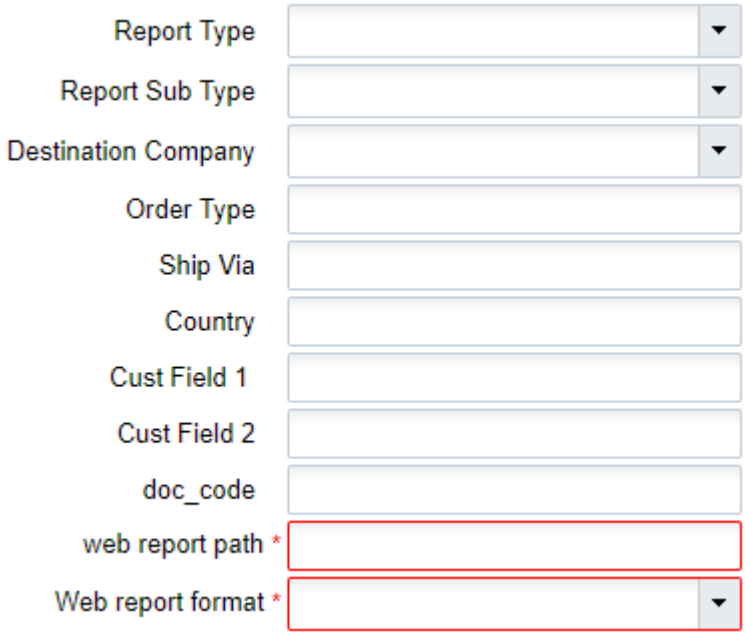
1. From the **Company Report Type** tab, all the report types will have a pre-configured document assigned, which you can modify to meet your operational requirements.



2. To add a new Report Type, click **Create**, located on the right-hand side corner.



3. The **Create** button opens a sub-menu that allows you to configure a new Company Report.



4. In the following example we are using the **Order Packing** Report Type:

Report Type
BOL
BULK_PICK_REPORT
COMM_INV
ESPL
ESPL
INBOUND_RECEIPT_RE...
OBLPN_PACKING
ORDER_PACKING
PALLET_PACKING

Pre-established Templates

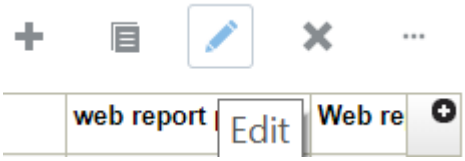
Company Report Type has some valid pre-established templates that are also pre-configured. However, if your current operational flow does not require any additional configurations, then you do not need to further configure the report.

If you need to do additional configuration, complete the following steps:

1. Select one of the available report types.

Report Type
BOL
BULK_PICK_REPORT
COMM_INV
ESPL
ESPL
INBOUND_RECEIPT_RE...
OBLPN_PACKING
ORDER_PACKING
PALLET_PACKING

2. Click **Edit** to modify the selected **Report Type** options.



3. The **Edit** menu opens with the following configuration. (The current configuration will allow the report to be called for any company, and requires no additional configuration).

4. The following is an example of an **Order Packing Slip** designed with specific information. If the report is using an internal template, doc_code can be left blank.

Report Type	ORDER_PACKING
Report Sub Type	_generic
Destination Company	*
Order Type	
Ship Via	
Country	
Cust Field 1	
Cust Field 2	
doc_code	
web report path *	
Web report format *	

Calling Label Designer Templates

Company Report Type can summon any Label Designer template that was created from scratch using that tool. Additional configuration is also available if your current operational flow requires it. Follow the above steps if you need to create or modify a current report.

Complete the following steps for additional configuration:

1. In this example, once the **doc_code** field has a value that matches an actual Label Template Code, then the system will not take into account the **Report Sub Type** value.

Report Type	ORDER_PACKING
Report Sub Type	_generic
Destination Company	COM123
Order Type	DUAL
Ship Via	INTER
Country	UG
Cust Field 1	
Cust Field 2	
doc_code	test_func
web report path *	
Web report format *	

Calling Web Reports Templates

Company Report Type can summon any Web Reports template that was created from scratch using that tool. Additional configuration is also available if your current operational flow requires it. Follow the above steps if you need to create or modify a current report.

Complete the following steps for additional configuration:

1. In this example, once the **Report Sub Type** has the WebReport value, the system allows you to use the last two fields ('web report' path and 'Web report format').

Report Type	ORDER_PACKING
Report Sub Type	WebReport
Destination Company	COM123
Order Type	DUAL
Ship Via	INTER
Country	UG
Cust Field 1	
Cust Field 2	
doc_code	
web report path *	Support/Oracle/Order Packing Test
Web report format *	

xls
pdf
csv

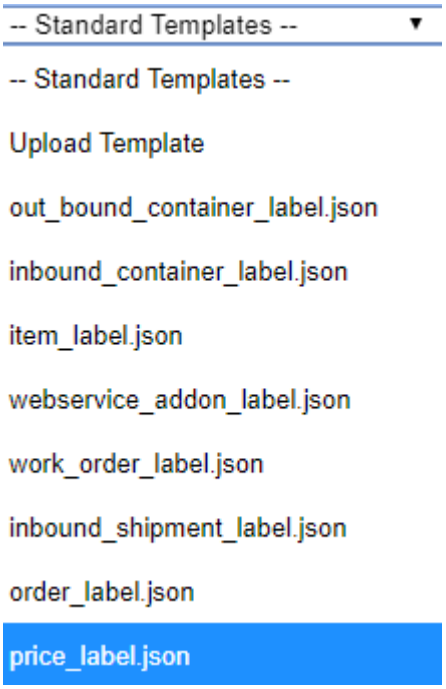
4 Price Labels

Price Labels

Label Designer allows you to configure settings for Price Labels, the same way that you can configure settings for Labels and Documents.

To configure settings for Price Labels, complete the following steps:

1. Select a **Price Label** template from the **Standard Templates** drop-down:



2. Configure your Price Label settings via the **Edit Document Label** pane just as you would for other types of Labels:

Note: You cannot configure multiple pages for price labels. You can only configure a single page.

Editing Document and Label Pages

Once you have saved your Label/Document changes, you can review and update your page settings.

Page settings will automatically be set to default settings which are the following:

Label Type	Default Dots Per Inch (DPI)	Default Page Size Type
Label	203	4X6 Portrait
Document	203	No Default, select desired Portrait or Landscape size.

You can view your page template in **Grid** view if you like, via Imperial or Metric measurements. You can also do your own width and height calculations for customized paper size if you need to.

Anything higher than 300 DPI is not recommended as it can cost more to print.

To make changes, change the value you want to change and then click save.

Note about Label DPI Zebra Printers: 203 DPI labels are the same labels that you use to print 300 DPI labels. The difference is that if the printer is configured to print 203 or 300 DPI. You need to print to a printer that is setup for 300 DPI printing for the 300 DPI label templates and a different printer for 203 DPI label templates. Otherwise, you will have to reconfigure your printer between printing each time. Document Label Designer has no problem printing any resolution size. If you are experiencing any issues, check your printer/template settings and configuration.

Document Page Definitions

Label Line Item	Description
Creating A Page	To create a new Page, click the Page at the top of screen.
Edit A Page	To bring up the edit pane, click the paper on your page.
Dots Per Inch	How many dots in an inch of ink or laser burning. Also known as DPI.
Width	Calculated by deciding how many inches a document is wide and then multiplying the number of inches by the dots per inch in order to find out how many dots wide a page will be printed. If you need to do this in metric then an inch is 2.54 cm. For 203 dpi one 1 mm is 8 dots and for 300 dpi 1 mm is 10 dots.
Page Type	Allows you to select if this is a First Page, Middle Page, or Last Page in this template. You can only have one First Page, Middle Page, or Last Page. You must always have a First Page in your template.
Page Size Type	Allows you to select what type of page size you want by default. This will auto calculate your width and height based on your Dots Per Inch. Not a required field.
Document	Allows you to select what document or label you want to apply the page to.
Canvas Zoom	Allows you to zoom the whole page in and out to better see your document or label. The canvas is like digital paper and it represents the document or label. Canvas zoom can help you view items closer or further away so that you can move elements around on the page or see how the page is looking as a whole.
Disabling Buttons While Loading	The buttons at the top making a new page, document, text, barcode, box, or line. Buttons are grayed out and disabled while it is loading.

5 Creating Barcodes

Creating Barcodes

Label Designer allows you to create 1D and 2D barcodes. You can create static and dynamic barcodes.

Note: WMS does not currently support static and dynamic repeating barcodes.

To create a barcode, complete the following steps:

1. Select an option from the Font Type drop-down to create a barcode for a zebra printer.

   If you leave the Font Type blank, the system will generate an Oracle barcode. Oracle barcode sizes are consistent with the height and width values that you set.

   Note: Barcodes for zebra printers may not scale automatically to the height and width values that you set. Sometimes zebra printers may recalibrate which can affect the scaling. You may need to make adjustments to the template over time to make sure your barcode is the size that you need. Zebra barcodes are recommended if you are printing a high volume of barcodes at the same time as they will print faster.

2. When you click on a barcode, the **Edit Barcode** pane appears on the right, letting you know you are editing a barcode.

3. To create a new Barcode, select the label you want from the **User Defined Templates** drop-down and then click the **Barcode** option at the top of the screen.

For document barcodes, your font type will be for 1D barcodes only. Document barcodes will use a font instead of an image-based barcode. This is fast to generate and creates small .pdfs which will download faster.

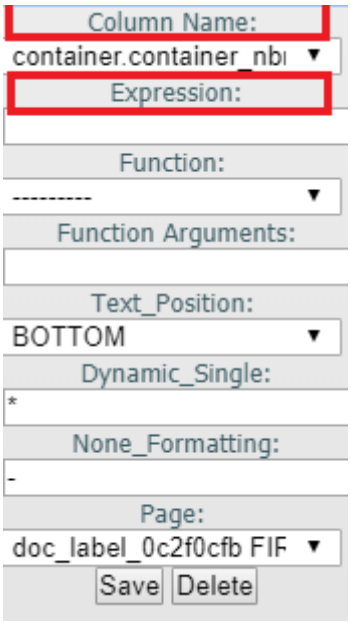
You are limited to 39 characters on a 128 width type barcode. You can go beyond this width on image-based 1D barcodes.

Note: Your scanner may not be able to exceed these specifications.

In most cases, you are going to want to select dynamic as the data type for your barcode. You can use static in cases where you have to repeat the same data. Your data will change for dynamic bar codes. Your static data will need an assigned value and you can only choose value specifications that exist within the barcode. Your dynamic data will map to certain values.

A common example of what you may map to would be the column name / container number. You should only specify one field.

4. Specify any **Function** and **Function Arguments** you have for facility time or wave time. For example, this allows you to specify formatting and coordinate date and time for your waves.



5. Once you have specified all of your barcode settings, click **Save**. The system will let you know if there are any errors.

See the *Expressions* section for more details about creating Expressions.

The most common barcode settings you will use are **Column Name** or **Expression**.

Barcode Definitions

Barcode Field	Description
Move Barcode	You can move barcode by clicking on it and then dragging the barcode to where you want it to be positioned. The barcode will automatically save its position when you are done dragging. Another way to move barcode is by entering its x and y coordinates. X = 0 and Y = 0 is the top left corner of the page. The bottom right corner are X = page width and Y = page height. So by entering the number you want for positioning it allows you to put the barcode where you want it. After you type a number and then hit tab or click out of the edit box, the barcode will move to that location. In order to save your change you have to click the save button.
Type	Allows for the barcode type that you want to use.
Data Type	Allows you to set your data to 1 of 3 values. Static means your text will never change like a title but this still allows for literals to be used for different languages. Only the Value field can be used with Static Data Types. Dynamic means you will only have one value that changes depending on data. An example would be an order header or order number on an order.

Barcode Field	Description
	Dynamic Repeating is a means you can get back more than one result typically used in a table of values. Example would be like order detail where the order quantity is being displayed for an order. You could get back more than one order detail for a single order. Column Name, Expressions, and Functions can be used with Dynamic Data Type. Column Name, and Functions can be used with Dynamic Repeating Data Type.
Font Type	If set to blank, an image based barcode is generated for PDFs. If set to a barcode font for a pdf, this will use a barcode font size which allows for barcodes to be generated faster than images. Note barcode fonts are only possible for 1D Barcodes. For labels fonts are ignored for 7.0.1 and below. In 7.0.2 and up font based barcodes set to any font will do the standard Zebra barcodes. Barcode widths since can dynamically change based on the calibration of the printer. If you are experiencing problems with this issue, you can try updating the firmware on your Zebra printer. So we now have image based barcodes for labels too as of 7.0.2. By un-setting the barcode font image based barcodes will be generated. These will be the precise size as in the label designer. These print more clearly in landscape and portrait mode.
Value	Allows for a static unchanging text good for things like the title of a column etc.
Column Name	Allows you to map a particular field from a table or standard functions on a table to a Dynamic text values in the barcode. Dynamic Repeating is currently not supported and would require a lot of work to support it. The only label type that bridges three different database tables and lets you print them with one set of mappings magically without failing is the IB container label type. Since containers do not always exist for IB shipment and IB shipment details, you cannot use a container for the mapping to get to the container number or any other container field. You need to use the IB shipment detail container number mapping for container numbers. It is the only one that is safe to use for the same label across all three screens. Never use a container number for an IB container label. Use the IB shipment detail container number. Use IB shipment and IB shipment detail fields for your mappings for an IB container label. In the case of an IB shipment in transit status, the IB containers do not exist in the system so there is no way to map these fields and you will crash the label. You must test configuration of labels in PreProd or in UAT.
Function	Allows for a standard or custom function that can be mapped to Dynamic or Dynamic Repeating Data Type.
Function Arguments	Allows for addition values to be passed to a standard or custom function. The function must already be developed to handle certain arguments in order to process the data being passed to it. For Example. FACILITY_TIME function can have arguments like the following:

Barcode Field	Description
	Passing Function Parameters: Example Of Date Time Formatting: **%d/%m/%Y - %H:%M:%S** Time Format Reference: *http://strftime.org/*
Expression	This can only be used for Dynamic Data Type and the Dynamic Single needs to be set to first otherwise an error can occur if more than one result gets returned.
Dynamic Single	When barcode is set to Dynamic for the Data Type and more than one result comes back for that field, only one thing can be displayed in the barcode. So then the default value for dynamic single of ***** is displayed. This can be changed to another value to show that more than one result is getting returned and so it cannot be displayed because it will not fit on the page. The other option is to set Dynamic Single to the word ***first*** . If Dynamic Single is set to first, only the first result is shown. This works well when you know you will get a bunch of the same results for this particular look. For instance, if you only expect a single order and you have a field mapped to order header order number, then setting Dynamic Single to first makes sense because you only need to display one result. However, if for a single order the field was mapped to order detail order quantity, then we expect to get back more than one result. If we do then we need to show a * or change the Data Type to Dynamic Repeating and print multiple pages / labels to show all of these details.
X	Position the Barcode on the X-Axis based on the number of dots of ink or thermal burning on a page.
Y	Position the Barcode on the Y-Axis based on the number of dots of ink or thermal burning on a page.
Width	How wide your barcode is. New for labels in 7.0.1 auto scaling barcodes will scale up or down mathematically depending on the amount of data and the width defined in the barcode layout. Barcodes already have a set width in pdfs. Auto scaling barcodes are only possible for Code 128, Code 128A, Code 128B, Code 128C, Code 128 UUC, and Code 39. Other barcode formats failed to scale or errored out when auto scaling.
Height	How tall your barcode is.

Barcode Field	Description
Rotate	Rotates the barcode. For Labels you can rotate by 0, 90, 180, 270 degrees as valid rotations. For Documents PDFs you can rotate between 0 - 90 degrees.
None Mapping	When there is an empty value also known as None what do you want to display? Default display value is - Note this is not the same thing as 0 this means no value at all.
Text Position	Show no text with the barcode, show the text value below the barcode, or show the text value above the barcode.
Page	Set what page this barcode will appear on.
Disabling Buttons While Loading:	The buttons at the top making a new page, document, text, barcode, box, or line, etc. Are grayed out and disabled while it is loading.
Special Mapping Considerations:	IB Shipments from an IB Container or Allocation. For doing an IB Shipment from an allocation we cannot promise data will always be present from an IB container through the received shipment foreign key on the from container foreign key on an allocation. The reason for this is that IB containers can be combined for instance in which case we lose the IB shipment reference. There are other situations that arise that can also cause this too.
Adjusting Barcode Width To Fit On Page With Zebra Printer Settings:	Update Firmware: Zebra recommends updating the firmware of the Zebra printer to the latest version to fix the barcode changing size issue. Adjusting Unprintable Area Right: *https://support.vendhq.com/hc/en-us* Adjusting Print Width: This can be adjusted by hand or by doing a calibration of the label stock. **Note:** Refer to the *Zebra Support* page for more information. Adjusting Max Allowed

Create a Label or Document with Dynamic Repeating Barcodes

You can print labels more efficiently using Dynamic Repeating Barcodes paired with Dynamic Repeating Text with DPI Dimensions.

A.Follow these *steps* to create your desired label/document.

Input Values for Text Elements in a Label

Now, to input values for the text elements in a label, do the following:

1. If you want configure the height and length of the text elements, disable the “DPI Dimensions” flag.

   > **Note:** By default, the DPI Dimensions flag remains unchecked/disabled.

   Here, you can enter the values for “Length” and “Height” via the respective exposed fields on the Edit Text pane. Alternately, you can also drag to resize the text box on the canvas.
2. However, if you wish to configure the DPI dimensions of the text elements, click on the checkbox for the “DPI Dimensions” flag.

   Here, you can enter the values for “DPI Width” and “DPI Height” via the respective exposed fields on the Edit Text pane. Alternately, you can also drag to resize the text box on the canvas.

B.Follow these *steps* to pair dynamic repeating barcode with dynamic repeating text.

Use a Dynamic Grid DPI

Now, to make use of a barcode grid DPI, do the following:

1. Click on the desired barcode on the label designer canvas.
2. Go to the “Edit Barcode” pane, select your desired barcode type via the “Type” drop-down.
3. On the Data Type dropdown, select “Dynamic Grid”.
4. Enter the values for ‘X’ and ‘Y’ coordinates to input the grid position.
5. Enter the values for “Height” and “Width” on their respective fields to define the grid dimensions.

Detailed Barcode Editing

Document Label Designer allows you to turn off dynamic scaling for font based barcodes, so that you have an extra option to fine tune barcode sizes. You also now have the option to

toggle to a view that renders/prints sample data.

After you select your barcode and you are using the **Edit Barcode** pane, the following fields are now available to assist you with more detailed barcode editing:

Edit Barcode Field	Function
Dynamic scaling	Allows you to do dynamic scaling for Zebra text-based label barcodes versus fixed sizes. **Note:** when you turn off Dynamic Scaling, the barcode is a fixed barcode size and will not resize with the data.
Sample data	Displays sample data instead of the default value from value, column name, function, expression, or page number.
Sample data as expression	Test expressions for print preview to test the template.

Print Preview

You can now actively test out your expressions in print preview without having to print a label while you are designing your label template.

Once you have selected a barcode or text element, you can use the **Sample data** option in **Edit Barcode** and put in the data you want to seefor print preview**.** This sample data is a useful feature if you are setting up a new warehouse and you don't have any existing data.

> **Note:** Currently print preview does not work for languages (such as Arabic) that read right to left. For these languages, print your label to confirm correctness.

If you use the **Sample data as expression** option, then you can use the same data to workout your expressions however, there is no data lookup in the sample data for the print preview because there is no existing data.

> **Note:** expressions need to be test written with the data already looked up in the expression

For example:

Instead of adding "\~container.container_nbr" in a data expression like you can in a real expression, you should add "LPN000001."

6 Multi-Field Barcode Scanning

Multi-Field Barcode Scanning

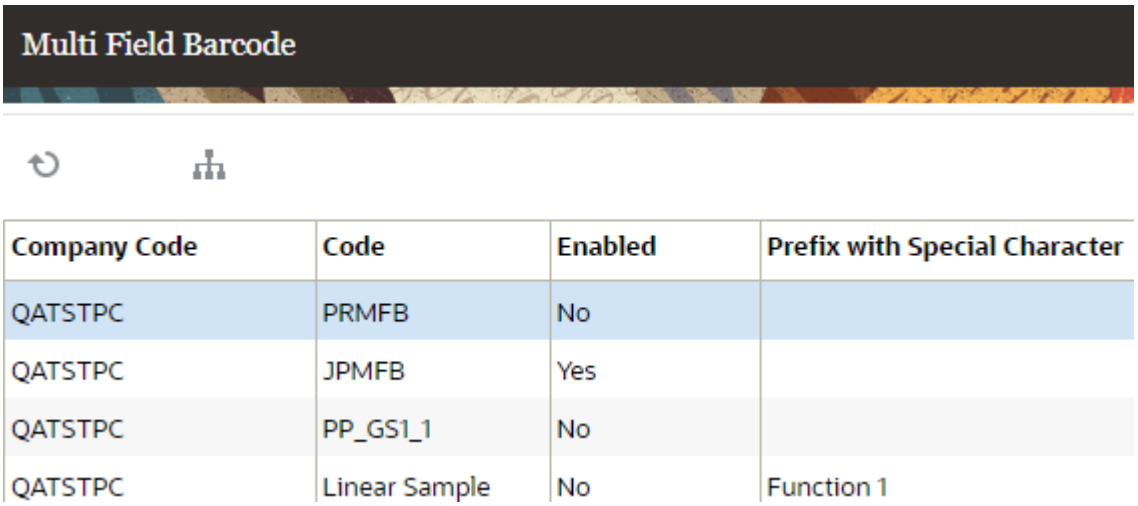
You can now generate/print valid Multi-field barcode scanning using Label Design. This new functionality allows you to specify the barcode construction by editing the expression field of the barcode on Label Design. This gives you the ability to have a function's text inside the expression field, so that the system can generate a multi-field barcode with the different parts of the GS1 code. You can use either Data Matrix type or UCC_128 to create this barcode, and you should map the Article ID to an existing column, and use the following format when creating a label expression:

[Start C Character] + [FNC1 Character] + Barcode Prefix + [Code A Character] + Merchant ID + [Code C Character] + Article ID (without Merchant ID + [Symbol Check Character] + [Sop Character]

Multi-field Barcode Setup

Multi-field barcodes such as those that are part of the *GS1 specification* encode multiple fields such as LPN, Item, and expiration date into one barcode. This barcode may be a 1-D format (eg. GS1-128) or 2-D format (DataBar). 2-D formats are capable of storing more characters and therefore more fields within one barcode.

From the **Multi Field Barcode** tab in Oracle Warehouse Management (WMS) Cloud, you can configure multiple classes of multi-field barcode scanning and define different sets of supported fields within each class.



Company Code	Code	Enabled	Prefix with Special Character
QATSTPC	PRMFB	No	
QATSTPC	JPMFB	Yes	
QATSTPC	PP_GS1_1	No	
QATSTPC	Linear Sample	No	Function 1

Each multi-field barcode class defines all possible fields, and not the contents of any one specific barcode. A class is identified by different prefixes. The barcode scanner must be configured appropriately. Multiple classes can be defined and enabled or disabled. The application has a default multi-field barcode class (MFB1 Sample) that is a subset of the GS1 spec, including fields (AI's or application identifiers in GS1 terminology) that are frequently used in warehouse management. Each field is associated with a standard barcode type.

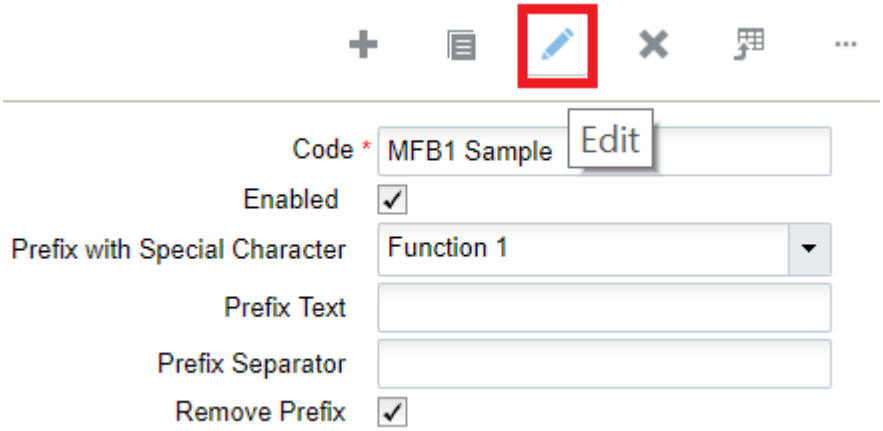
Working with Multi-Field Barcode Classes (Application Identifiers)

The following few sections review steps for working with Multi Field Barcode Classes. To define your data setup, you will be using a delimiter and you will use expressions in Document Label Designer to compose them. Your Multi Field Barcode configuration will define how the data gets broken up from the text that gets scanned in from the barcode.

> **Note:** there are many possible GS1 configurations. This is one example. Please refer to the *GS1 Manual* for the configuration you need and determine what your mappings should be. Also refer to the Barcode Construction Example in the *Support for FNC Characters for GS1 Barcodes* topic for more examples and details.

Multiple classes can be defined and enabled or disabled. To edit barcode details such as enabling a class or define a prefix with special character, complete the following steps:

1. From the Multi Field Barcode screen, select the barcode you would like to edit.
2. Click **Edit**. Edit the barcode details and then click **Save**.



You can also click **Create** to add a new Multi Field Barcode and specify details.

To edit Multi Field Barcode Class fields, complete the following steps:

1. From the Multi Field Barcode tab, select the multi field barcode you would like to edit.
2. Click **Details**.

3. Select the Field you would like to edit. In the following example, the Variable Count field is selected.

Multi Field Barcode » Multi Field Barcode Class Field

Multi Field Barcod	Field Identifier	Field Name
QATSTPC/MFB...	79	IB Container
QATSTPC/MFB...	17	EXPIRY DATE (YYMM...
QATSTPC/MFB...	37	COUNT
QATSTPC/MFB...	30	VARIABLE COUNT

4. Edit the Multi Field Barcode Class details and click **Save**

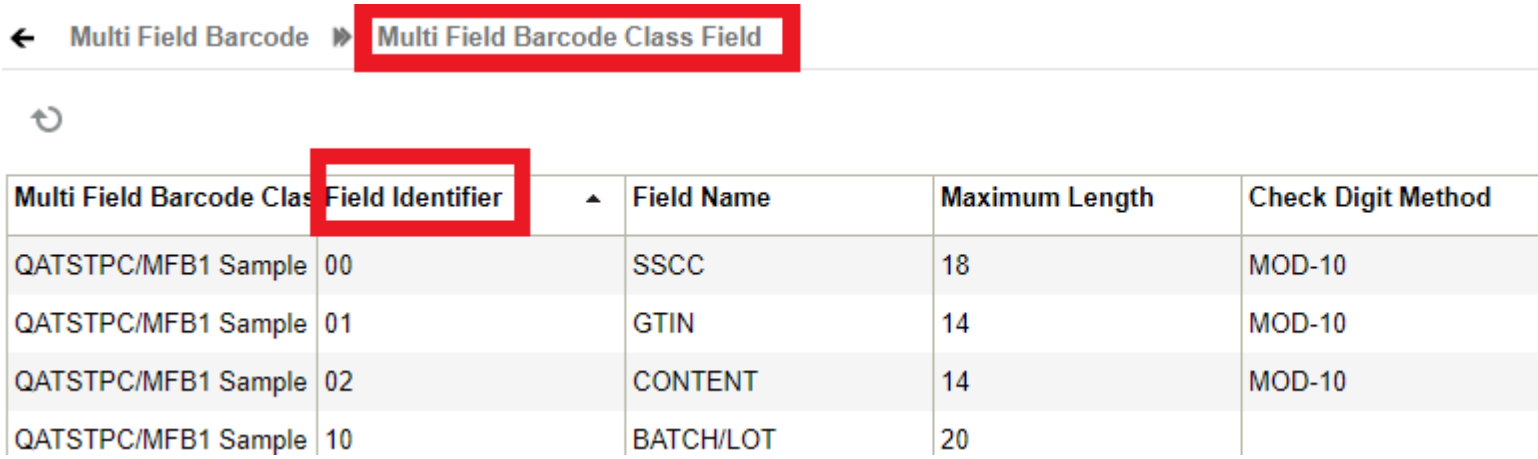
Multi Field Barcode Class	QATSTPC/MFB1 Sample
Field Identifier	30
Field Name	VARIABLE COUNT
Maximum Length	8
Check Digit Method	MOD-10
Barcode Type	QATSTPC/Count
Numeric Flag	☑
Fixed Width	☐
Delimiter	Group Separator
Delimiter Text	
Strip Check Digit	☐
Date Format	YYMMDD

The application has a default multi-field barcode class that is a subset of the GS1 spec, including fields (AI's or application identifiers in GS1 terminology) that are frequently used in warehouse management. Each field is associated with a standard barcode type.

Sl No	Column Name	Description
1	Company Code	Company Code to which the barcode is defined
2	Code	Code for the multi-field barcode
3	Enabled	User can enable or disable the multi-field barcode using this flag. By Default the flag is set to false
4	Prefix with special Character	User can select a prefix with special character from this drop-down, including Function 1 or Group Separator.
5	Prefix Text	You can define the prefix for the multi-field barcode. This prefix enables the system to automaticatically identify the scanned data as a multi-field barcode based on the text that has been prefixed in the multi-fied barcode. You can use any alphabets or a combination of alphabets and special characters supported by GS1 as prefix text. For example: **]C1** - It is the GS1 Data Bar extended. But there are flavors according to the Laser Gun and design of the Barcode. For example]C3 **]Q1** - It is the GS1 QR Code - But there are flavors according to the Laser Gun and design of the Barcode. For example]Q3]d1 It is the GS1 Data Matrix – But for all the different real barcodes we have collected, no one have this code. We have always, in particular for pharmaceutical products the flavor]d2
6	Prefix Seperator	If you want to define multiple prefix text, you can define it in the “Prefix Text” column seperated by a special character. Inorder for the system to identify the special character, user can define the special character in this column. You can use any special characters supported by GS1 as prefix text.
7	Remove Prefix	If you to remove the prefix from the mult-ifield barcode, you can enable the flag. By default the flag is set to false

Individual fields within a multi-field barcode are prefixed by a code (Field Identifier) which in GS1 is a two digit (or more) number, followed by the value of the field. The field may be fixed with or variable field. If it is variable, it is ended with a delimiter.

From the **Multi Field Barcode** tab, click details to view the **Multi Field Barcode Class Field**.



The following table provides details for Multi-Field Barcode field definitions, including WMS Barcode Types that are supported:

Sl No	Column Name	Description
1	Field Identifier	Individual fields within a multi-field barcode are prefixed by a code. User can define that code as field identifier
2	Field Name	Field name
3	Maximum Length	Max length of the field, not including field name but including check digit.
4	Check Digit Method	The last digit of all fixed-length, numeric GS1 Identification Keys is a check digit that ensures the integrity of the key. All *GS1 ID Keys* need a check digit, except Component/Part Identifier (CPID), Global Individual Asset Identifier (GIAI) and Global Model Number (GMN).
5	Barcode Type	The following are the barcode types supported: • IB Container • OB Container • Pallet • Item • IB Shipment • Location

Sl No	Column Name	Description
		• Cart • Batch Nbr • Asset • Asset Seal • Serial Number • OB Load • Inv Attribute A • Inv Attribute B • Inv Attribute C • Inv Attribute D • Inv Attribute E • PO Nbr • IB Trailer • OB Trailer • Seal Number • Shipping Container • Expiry Date • Count • Weight • Manufacture
6	Numeric Flag	To define whether the field is numeric or not. If the field is number, then user can enable the flag Default False.
7	Fixed Width	To define whether the field length is fixed or not. If the field length is fixed, then user can enable the flag Mutually exclusive with delimiter_id/text. Default False.
8	Delimiter	Delimiters are used to specify the boundaries between separate regions in a scanned data string. When you are entering delimiters, note that the Separation Delimiter field is required, but the Start Delimiter and End Delimiter fields are optional.
9	Delimiter Text	If the field is variable length, user should define delimiter text so that system can identify the field from the multi-field barcode
10	Strip Check Digit	Some fields have a check digit defined. This check box applies only in these cases and it will remove the last digit from that field and only

Sl No	Column Name	Description
		use the remaining for processing. You may want to define a check digit depending on how you are using the values in the fields.
11	Date Format	Many different date formats are supported. The following are some examples of date formats: MMDDYY as %m%d%Y example output: 09012015 MM-DD-YY as %m-%d-%Y example output: 09-01-2015 MM/DD/YY as %m/%d/%Y example output: 09/01/2015 DD/MM/YY as %d/%m/%Y example output: 01/09/2015 DD MM YY as %A %d %B %Y example output: Monday 01 September 2015 Refer to the following url for a full listing of option characters for date and time formatting: *http://strftime.org/*

Note: WMS currently only supports one serial number for multi-field barcodes. WMS does not currently support multiple serial numbers for multi-field barcodes.

All RF screens can now recognize multi-field barcodes based on the prefix.

For example:]C101IBLPNRGN2951800102RGITM003~^~0310~^~

- Once a certain class is detected, it extracts all the component fields and they are used within the current RF operation. For example, if a receiving option expects an LPN number, manufacturing date, SKU, and count via entry fields (that may be on multiple screens), and a barcode containing the LPN number, manufacturing date and SKU is scanned, all three values will be extracted and used automatically.
- The application validates all three fields automatically and moves to the next scan.
- A new RF parameter ***mfb-restrict-to-curr-fld*** has been added to certain RF modules like Cycle Count, which can be configured to restrict multi-field barcodes to just apply to the current field and not to extract all fields:

Support for FNC Characters for GS1 Barcodes

Label designer allows you to have certain special characters in font-based Gs1 Linear and Datamatrix barcodes. You can select a value for font type in label designer while editing a barcode.

GS1 font-based barcode functionality allows you to:

1. Choose the type of barcode field you need (Code 128, A B or C).
2. Decide on the fields that make up the barcode.
3. Bring together the fields in an expression that places the FNC characters as required.

Functions

The following functions can be used for this:

gs1_li_code - Accepts a keyword and generates a ZPL instruction that results in this special character on the label. *This should be used for GS1 linear barcodes*. Supported keywords are: FNC1, FNC2, FNC3, FNC4, STARTA, STARTB, STARTC, CODEA, CODEB and CODEC. Note - Every keyword is NOT applicable to every subset (A, B and C), so the users must know what kind of combinations they are using.

Sample expression to generate an FNC1 character: gs1_li_code('FNC1')

gs1_dm_code - This is the GS1 - Datamatrix equivalent of gs1_li_code. Supported keywords are 'FNC1', 'FNC2', 'FNC3'.

Sample use: gs1_dm_code('FNC1')

chr - This is a basic function. Given an integer value, it returns the corresponding ASCII code. You should ensure that sensible data is passed in based on the column mapping used.

Sample use: chr(29)

gs - Helper function which returns the group separator character (ASCII code 29). This is merely for convenience but it can also be generated using chr (as in the example above).

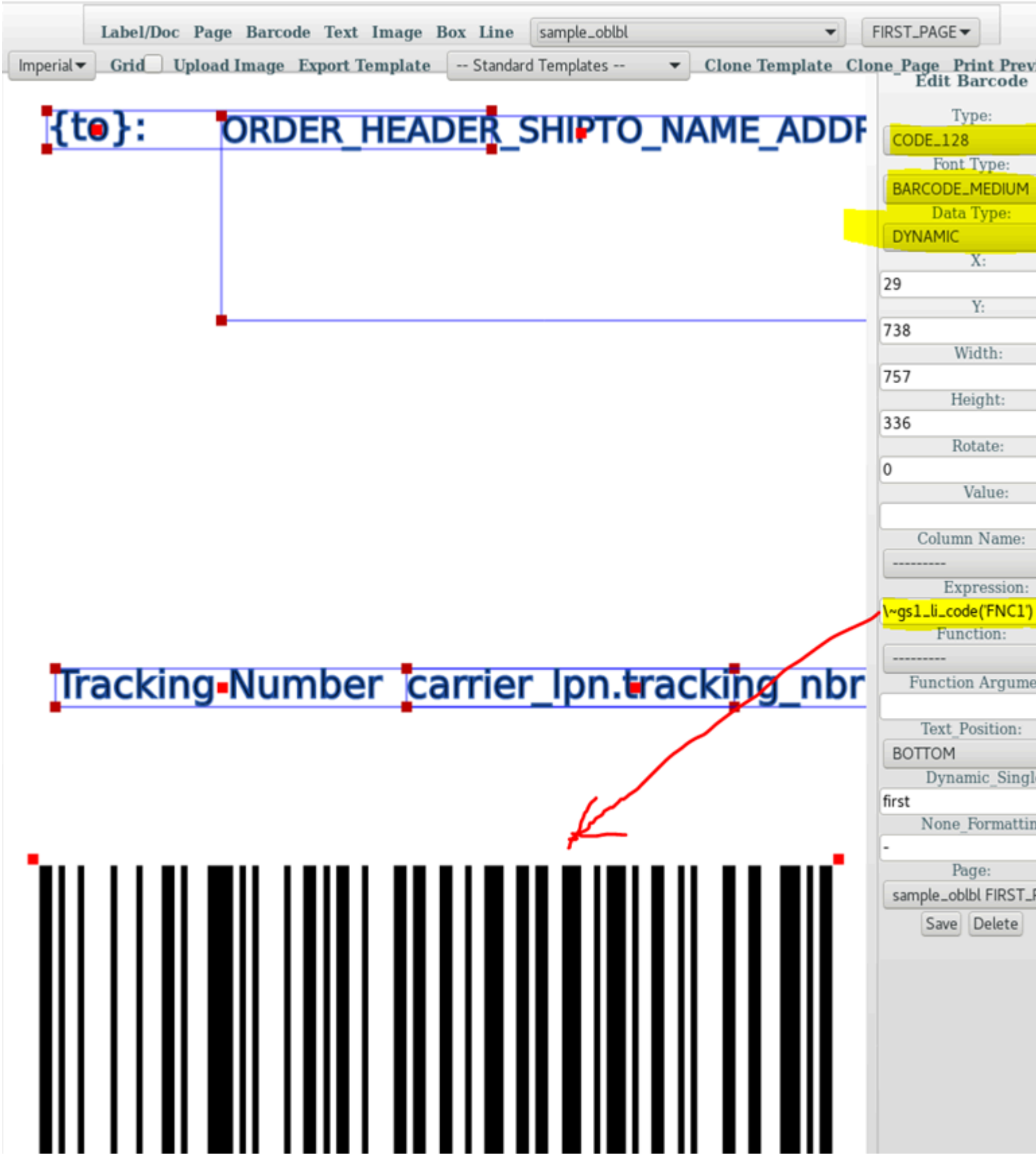
Commonly known issues/recommendations

1. Sometimes all that is required is to use Subset A, B or C.
2. Missing quotes or double quotes, extra quotes, using ` instead of '. Because of the flexibility and power of expressions we have had to adopt different syntax to support different things.

Example:

In the following Expressions example, pay close attention to the way single quotes and double quotes are used. The 00 is an example of an Application Identifier located at a strategic place (if it is to be used).

```
gs1_li_code('FNC1') + "00" + "\~container.container_nbr"
```


Label/Doc Page Barcode Text Image Box Line
sample_oblbl
FIRST_PAGE
Imperial
Grid
Upload Image
Export Template
-- Standard Templates --
Clone Template
Clone Page
Print Prev
Edit Barcode
{to}:
ORDER_HEADER_SHIPTO_NAME_ADDF
Type:
CODE_128
Font Type:
BARCODE_MEDIUM
Data Type:
DYNAMIC
X:
29
Y:
738
Width:
757
Height:
336
Rotate:
0
Value:
Column Name:
Expression:
\~gs1_li_code('FNC1')
Function:
Function Argume
Text_Position:
BOTTOM
Dynamic_Singl
first
None_Formattin
Page:
sample_oblbl FIRST_F
Save
Delete
Tracking Number
carrier_lpn.tracking_nbr

Barcode Construction Example

Note: there are many possible GS1 configurations. This is one example. Please refer to the *GS1 Manual* for the configuration you need and determine what your mappings should be:

[Start C Character] + [FNC1 Character] + Barcode Prefix + [Code A Character] + Merchant ID + [Code C Character] + Article ID(without Merchant ID + [Symbol Check Character] + [Sop Character]

Article ID: XYZ028373401001860208

This comes back to us as full string but can be broken down by the following components:

- Merchant ID: XYZ
- Consignment ID: 0283734
- Article count: 01
- Product Code: 00186
- Service Code: 02
- Paid Postage Indicator: 0
- Check Digit: 8

Barcode Prefix = The items listed below in between the Function 1 Symbol & Code A character. This is a fixed value that will always be the same for this example. In this example the value is "0523349852737927562".

Barcode Prefix = [Application Identifier][Indicator][Company Prefix][Item Reference][Check Digit][Application Identifer]

Merchant ID = This is a fixed 3 digit value, in this example it is XYZ.

Article ID (without Merchant ID) = The Article ID defined above needs to have the first three characters and the merchant ID stripped because there needs to be a Code C Character in between the Merchant ID and the other components of the Article ID.

Article ID = [Consignment ID][Article count][Product Code][Service Code][Paid Postage Indicator][Check Digit]

You would then build the barcode like the following example:

[Start C Character] + [FNC1 Character] + 0199312650999998911 + [Code A Character] + XYZ + [Code C Character] + Bytes 4-27 of the LPN Tracking number + [Symbol Check Character] + [Stop Character].

gs1_li_code (GS1 Linear barcode) – This takes in a keyword and returns a special ZPL instruction that results in this character on the label. This should be used for GS1 linear barcodes. Supported keywords are: FNC1, FNC2, FNC3, FNC4, STARTA, STARTB, STARTC, CODEA, CODEB and CODEC.

Note: every keyword is NOT applicable to every subset (A, B and C).

Sample use: gs1_li_code('FNC1')

gs1_dm_code(GS1 Datamatrix barcode) - This is the Datamatrix equivalent of gs1_li_code. Supported keywords are 'FNC1', 'FNC2', 'FNC3'.

Sample use: gs1_dm_code('FNC1')

chr - This is a basic function. Given an integer value, it returns the corresponding ascii code. You should ensure junk values are not passed in gs. This is a helper function which returns the group separator character (ASCII code 29).

Australian Post Example

The expressions are:

DataMatrix

```
"" if
 "\~carrier_lpn.tracking_nbr"== "" else gs1_dm_code('FNC1') + "0523349852737927562XYZ" +
 gs1_dm_code('FNC1') + "\~carrier_lpn.tracking_nbr"[3:27]
```

This expression and the dimension configurations together generate the ZPL instructions:

```
^FO640,100

^BXN,7,200,,,,_,1

^FD_10523349852737927562XYZ
 _1028373401001860208^FS
```

Linear Barcode (For barcode type 128)

```
"" if
 "\~carrier_lpn.tracking_nbr"== "" else gs1_li_code('STARTC') + gs1_li_code('FNC1') + "0523349852737927562" +
 gs1_li_code('CODEA') + "XYZ"
 + gs1_li_code('CODEC') + "\~carrier_lpn.tracking_nbr"[3:27]
```

This expression and the dimension configurations together produce the ZPL code:

```
^FO51,550

^BY3^BCN,175,N,N

^FD>;>80523349852737927562 >7XYZ>5028373401001860208^FS
```

Caution:

For Datamatrix the character _ (underscore) has a special meaning. This character specifies a ZPL instruction for the FNC1. If the data to be barcoded has _ already, then we may need to make some minor changes.

[Start C Character] + [FNC1 Character] + Barcode Prefix + [Code A Character] + Merchant ID + [Code C Character] + Article ID(without Merchant ID + [Symbol Check Character] + [Sop Character]

Image Based Barcode Support

Document Label Designer includes support for image-based barcodes and QR codes which can be used on labels and documents. This is helpful for GS1 labels and documents.

The following functions have been added to Document Label Designer:

- Python expressions for GS1 now work with image-based barcodes.
- Compatibility for Linear Barcode Code 128 and Data Matrix.
- AI Code Function support for ZInt, Zebra, and ID Automation.
- QR Code Support.

- Support for Linear And Data Matrix code.
- New GS1 Code function support that does not require you to have to choose a particular function for a
- particular output such as Zebra, ZInt, or ID Automation.
- Adjusted font scaling for pdf documents that are using the ID Automation font.

Document Label Designer includes an examples template called gs1_doc_sample for GS1 barcodes available from the standard template list. If you select this standard template, Label Designer creates a template with a whole set of example barcode expressions for GS1 and also examples for doing sample data expressions.

> **Note:** The underlining technology for image based barcodes does not support the gs1_code functions START A, B, C and FNC 1,2,3. If you want these particular codes, you need to use a font-based barcode. gs1_code, gs1_li_code, and gs1_dm_code functions are not compatible with image-based barcodes. Only gs1_ai functions are compatible with and work with image based barcodes.

The following are examples of expressions you can use for Linear and 2D barcodes:

Use this expression for Linear Barcode Code 128 Text Barcode:

```
gs1_code("FNC1") + gs1_ai("01") + "98898765432106" + gs1_ai("3202") + "012345" +
 gs1_ai ("15") + "991231"
```

Use this expression for 2D Barcode Data Matrix Text Barcode:

```
gs1_code("FNC1") + gs1_ai("01") + "98898765432106" + gs1_ai("3202") + "012345" +
 gs1_ai ("15") + "991231"
```

7 Document and Label Designer Expressions

Document and Label Designer Expressions

Expressions in Document Label Designer are python expressions that fit on one line. An expression has to be one line of code to be executed. With expressions you can do a lookup for a database table and field.

Expressions

With the new functionality, UI column name can be used in expressions to do a lookup for a database table and field. The following are examples of lookup:

```
u"\~carrier_lpn.tracking_nbr"
```

where carrier_lpn is the database table and the tracking_nbr is the field separated by a .

Two lookups separated by a space as a string:

```
u"\~order_header.custom_long_text_3
 \~order_header.custom_long_text_4"
```

The following is an example of writing an if condition statement: Note the "" around the field lookup so that the result is treated as a string.

```
"Not Manifested!" if "\~carrier_lpn.tracking_nbr " == "" else "\~carrier_lpn.tracking_nbr"
```

Slicing and length expression example: Note this example starts from the 3rd position to the end of the string, extracting that portion of the string.

```
"\~order_header.order_nbr "[slice(2,len( "\~order_header.order_nbr "))]
```

Split expression with accessing the first element example:

```
split( "\~order_header.custom_long_text_3 ",
 ";")[0]
```

Complex split expression example:

```
split("\~order_header.custom_long_text_3", ";")[0] + "\n" +
 split("\~order_header.custom_long_text_3", ";")[1] + " " +
 split("\~order_header.custom_long_text_3", ";")[2] + " " +
 split("\~order_header.custom_long_text_3", ";")[3] + "\n" +
 split("\~order_header.custom_long_text_3", ";")[4] + ", " +
 split("\~order_header.custom_long_text_3", ";")[5] + " " +
 split("\~order_header.custom_long_text_3", ";")[6]
```

The following is an example of a more complex inline if split expression:

```
str("(No \~order_header.customer_number)") if
 len(split("\~order_header.custom_field_5", ";")) < 12 else str("(No " +
```

```
split("\~order_header.custom_field_5", ";")[11] + "\~order_header.customer_number" +
")")
```

New Expressions Supported

```
u"str(5) + str(5)"

get_country_name("AD")

get_country_code("UNITED STATES")

get_us_state_name("GA")

get_us_state_code("GEORGIA")

"strftime(\~order_header.mod_ts, "%m/%Y")"
```

The following is an example of an expression for container weight:

```
str(int((10*\~container.weight)-0.5)+1).zfill(7) + "A"
```

A container weight of 65.46 would result in 0000655A. Zfill does the padding for the resulting string and int((10* \~container.weight)-0.5)+1 rounds the number up and moves up one decimal place.

To test your expression on your own computer, you will need simple eval python. You can look at more examples on the following website:

https://pypi.python.org/pypi/simpleeval

Passing Function Parameters

Example Of Date Time Formatting:

```
%d/%m/%Y - %H:%M:%S
```

See *Python strftime Directives* for more details on Date Time Formatting.

Blind Label Mappings:

```
"container.container_nbr"

"order_header.destination_company.code"

"order_header.dest_facility.code"

"company.code"

    dict_get(get_obj(),
     "order_header.destination_company.code")


    str(dict_get(get_obj(), "company.code"))[slice(5,len(dict_get(get_obj(),
     "company.code")))]
```

Python strftime Directives

Note: Examples are based on `datetime.datetime(2013, 9, 30, 7, 6, 5.)`

Code	Meaning	Example
`%a`	Weekday as locale's abbreviated name.	`Mon`
`%A`	Weekday as locale's full name.	`Monday`
`%w`	Weekday as a decimal number, where 0 is Sunday and 6 is Saturday.	`1`
`%d`	Day of the month as a zero-padded decimal number.	`30`
`%-d`	Day of the month as a decimal number. (Platform specific)	`30`
`%b`	Month as locale's abbreviated name.	`Sep`
`%B`	Month as locale's full name.	`September`
`%m`	Month as a zero-padded decimal number.	`09`
`%-m`	Month as a decimal number. (Platform specific)	`9`
`%y`	Year without century as a zero-padded decimal number.	`13`
`%Y`	Year with century as a decimal number.	`2013`
`%H`	Hour (24-hour clock) as a zero-padded decimal number.	`07`
`%-H`	Hour (24-hour clock) as a decimal number. (Platform specific)	`7`
`%I`	Hour (12-hour clock) as a zero-padded decimal number.	`07`
`%-I`	Hour (12-hour clock) as a decimal number. (Platform specific)	`7`
`%p`	Locale's equivalent of either AM or PM.	`AM`
`%M`	Minute as a zero-padded decimal number.	`06`

Code	Meaning	Example
`%-M`	Minute as a decimal number. (Platform specific)	`6`
`%S`	Second as a zero-padded decimal number.	`05`
`%-S`	Second as a decimal number. (Platform specific)	`5`
`%f`	Microsecond as a decimal number, zero-padded on the left.	`000000`
`%z`	UTC offset in the form +HHMM or -HHMM (empty string if the the object is naive).	
`%Z`	Time zone name (empty string if the object is naive).	
`%j`	Day of the year as a zero-padded decimal number.	`273`
`%-j`	Day of the year as a decimal number. (Platform specific)	`273`
`%U`	Week number of the year (Sunday as the first day of the week) as a zero padded decimal number. All days in a new year preceding the first Sunday are considered to be in week 0.	`39`
`%W`	Week number of the year (Monday as the first day of the week) as a decimal number. All days in a new year preceding the first Monday are considered to be in week 0.	`39`
`%c`	Locale's appropriate date and time representation.	`Mon Sep 30 07:06:05 2013`
`%x`	Locale's appropriate date representation.	`09/30/13`
`%X`	Locale's appropriate time representation.	`07:06:05`
`%%`	A literal '%' character.	%

Using Functions within Expressions

You can use functions within expressions in Document and Label Designer. The following are some examples:

Example of older method:

```
u"MYTEXT" + unicode(get_total_num_packed_containers_on_order(get_qs()))
```

Example of newer better method:

```
u"MYTEXT" + unicode(get_total_num_packed_containers_on_order(get_qs_list()))
```

The reason this is better is that some label types can have multiple querysets that should be passed into the function so get_qs might not be enough in some label templates.

Note: if your function has arguments you need to pass in you can do this as part of your expression:

```
u"MYTEXT" + unicode(get_total_num_packed_containers_on_order(get_qs_list(),
 args="MY_ARGS_GO_HERE"))
```

An example of passing actual arguments would be:

```
facility_time(get_qs_list(), args="d/%m/%Y - %H:%M")
```

Adjusting Timestamp using Expressions

You can use expressions to adjust the timestamp in your labels:

The following is an example of a received time stamp before trying to print it with specialized formatting:

```
strftime(\~container.received_time_stamp, "%d/%m/%Y %H:%M:%S") if
 \~container.received_time_stamp else "-"
```

Example of offsetting the date by 3 days earlier:

```
strftime(\~container.received_time_stamp - timedelta(days = 3), "%d/%m/%Y %H:%M:%S") if
 \~container.received_time_stamp else "-"
```

Example of offsetting the time by 3 hours earlier:

```
strftime(\~container.received_time_stamp - timedelta(hours = 3), "%d/%m/%Y %H:%M:%S") if
 \~container.received_time_stamp else "-"
```

Note: All expressions in document label designer must use **“ and not ‘.** If you need to have “ inside of another “ you can put a slash in front of the inner “.

Example:

```
“get_us_state_name(\"GA\")”
```

Most of the time this is not needed and there are other ways around it. For instance this could have also been written like the following.

Example:

```
str( get_us_state_name("GA") )
```

Testing Your Expressions

Install python IDLE from Ninite:

https://ninite.com/

```
functions={"slice": slice,

"len":
len,


"replace":str.replace,

"split":
str.split,

"splitlines":
str.splitlines,

"get_literal":
ast.literal_eval,

"dict_get":
dict.get,

"str":
str,

"int":
int,

"float":
float,

"unicode":
unicode,

"date":
datetime.date,

"time":
datetime.time,

"datetime":
datetime,

"strftime":
datetime.strftime,
```

```
"datetimenow":
datetime.now,

"strptime":
datetime.strptime,

"tznow":
tznow,

"get_obj":
self.get_obj,

"str_upper":
str.upper,

"str_lower":
str.lower,

"get_qs":
self.get_qs,

"chr" :
chr,

"gs" :
gs,

"gs1_li_code" :
gs1_li_code,

"gs1_dm_code" :
gs1_dm_code}
```

Note: the above functions do not get looked up with the \~ because they are built-in.

\~ is used for field look ups like column name or field functions. Additionally, standard functions get embedded so they do not need to be looked up.

If you are not able to install simple eval, you need to test on your local windows, mac, or linux computer with eval instead.

Note: you will need to look at the functions mapping to see how to map python commands to simple eval. For instance, "dict_get": dict.get, from above.

```
obj = {'container.container_nbr': 'x',

'order_header.destination_company.code': 'AE',

'order_header.dest_facility.code': '*',

'company.code': 'ACME'}
```

```
    expression = 'str(dict.get(obj, "company.code"))[slice(5, len(dict.get(obj,
     "company.code")))]'

print eval(expression)
```

You will need to swap back your expression afterwards so that it says the following:

```
    str(dict_get(get_obj(), "company.code"))[slice(5,len(dict_get(get_obj(),
     "company.code")))]
```

This allows it to work within the expression field in document label designer.

Another example:

```
obj = {'container.container_nbr': 'x',

'order_header.destination_company.code': 'AE',

'order_header.dest_facility.code': '*',

'company.code': 'ACME'}

    expression = '"" if str(dict.get(obj,
     "order_header.destination_company.code")) == "*" else str(dict.get(obj,
     "order_header.destination_company.code"))'

print eval(expression)
```

The expression that goes into document label designer is:

```
    "" if
     str(dict_get(get_obj(), "order_header.destination_company.code")) == "*" else
     str(dict_get(get_obj(), "order_header.destination_company.code"))
```

Another example of developing an expression in IDLE:

```
    expression = 'str("(No 000111)") if len(cust.split(";")) < 12 else str("(No " +
     cust.split(";")[11] + "000111" + ")")'

print eval(expression)
```

The expression that goes into document label designer is:

```
    str("(No \~order_header.customer_number)") if
     len(split("\~order_header.custom_field_5", ";")) < 12 else str("(No " +
     split("\~order_header.custom_field_5", ";")[11] + "\~order_header.customer_number" +
     ")")
```

8 Mappings

Mappings

The following mappings will help you understand what column names in the UI you can map for a particular label or document. You can also use mappings with expressions to assist you in determining what column names you can pull from for your label.

Mapping Details

Outbound LPN, Order, Task Allocated or Higher Status

'allocation', 'container', 'order header', 'order detail', 'order type','order status', 'carrier lpn', 'standard carrier', 'standard service service','standard service type', 'ship via', 'company', 'facility', 'item','payment method', 'lpn type', 'pallet', 'load', 'out bound stop','out bound stop detail', 'carrier', 'task', 'allocation run header', 'item', 'item characteristics', 'item metrics', 'item struct', 'item group definition','work order header', 'work order kit', 'work order component', 'work order status', 'work order type', 'work order activity type', 'in bound shipment', 'putaway type', 'inventory'

Order Less Than Allocated Status

'order header', 'company', 'facility', 'order status', 'order type', 'work order kit', 'work order component', 'order detail', 'item', 'item struct','item group definition', 'item characteristics', 'item metrics', 'item assignment type'

Location

'location', 'item', 'item characteristics', 'item metrics', 'item struct', 'item group definition', 'item assignment type', 'replenishment zone', 'task zone', 'inventory lock', 'location size type', 'location type', 'mhe system'

Item

'item', 'item characteristics','item metrics', 'item struct', 'item group definition'

Price Label

'price label'

In Bound Shipment

'in bound shipment', 'company', 'facility', 'load', 'inventory lock', 'in bound shipment detail','item', 'item characteristics','item metrics', 'purchase order header', 'purchase order detail','putaway type'

In Bound Shipment Detail

'in bound shipment detail', 'in bound shipment', 'item', 'item characteristics', 'item metrics','purchase order header', 'purchase order detail', 'putaway type', 'company', 'facility','load', 'inventory lock', 'inventory attribute', 'container in bound shipment detail cross reference', 'container','pallet', 'batch', 'inventory'

In Bound LPN

'container', 'in bound shipment', 'inventory', 'container in bound shipment detail cross reference', 'item', 'item characteristics','item metrics', 'inventory attribute', 'item struct', 'item group definition', 'harmonized tariff','uom', 'batch number', 'pallet', 'putaway type', 'asset', 'inventory lock', 'in bound shipment detail','purchase order header', 'purchase order detail', 'load'

Serial Number

'serial nbr', 'company', 'facility', 'item', 'item struct', 'item group definition', 'item characteristics','item metrics'

Serial Number Inventory

'serial nbr inventory', 'serial nbr', 'inventory', 'company', 'facility', 'item', 'item struct','item group definition', 'item characteristics', 'item metrics', 'inventory attribute'

Work Order Header

'work order header', 'work order kit', 'work order component', 'company', 'facility', 'item','item struct', 'item group definition', 'item characteristics','item metrics', 'work order status','work order type', 'work order activity type', 'inventory attribute', 'order header', 'order detail','location', 'uom', 'order status', 'batch number'

Inventory – UOM Case, UOM Pack

‘inventory’, ‘item’

9 Standard Functions

Standard Functions

This section includes standard WMS functions that are helpful for you to know when working with Label Designer:

Function Details

Function	Definition	Data Type	Argument
calc_container_cost_from_item **(qs, args =u")**	Calculate Container Cost From item **The sum of all the allocation packed quantity x inventory item unit cost for a container.**	Dynamic	None
calc_container_cost_from_order **(qs, args=u")**	**Calculate Container Cost From Order - The sum of all the allocation packed quantity x order detail** **cost for a container.**	Dynamic	None
company_name_address (qs, args=u")	**Company Address Fields - generates an address in the following format:** **company_facility_name company_facility_address company_facility_address2 company_facility_city, company_facility_state company_facility_zip company_facility_country**	Dynamic	None
container_mod_time(qs=None, args=u'%d/%m/%Y - %H:%M')	**Container Modified Time - The container modified time and date when the document or label is printed based on the container modified time zone** **location.**	Dynamic	**Date Format String.** ***http://strftime.org/*** **Default: "%d/%m/%Y - %H:%M"**

Function	Definition	Data Type	Argument
destination_company_name_address (qs, args=u")	**Order Destination Company Address Fields - generates an address in the following format:** **destination_company_name destination_company_address destination_company_address2 destination_company_city, destination_company_state destination_company_zip destination_company_country**	Dynamic	None
destination_facility_name_address (qs, args=u")	**Order Header Destination Facility Address Fields - generates an address in the following format:** **facility_name facility_address facility_address2 facility_city, facility_state facility_zip facility_country**	Dynamic	None
determine_sku_single_or_multi **(qs, args=u")**	**Determine Sku Single Or Multiple - looks at the iblpn to see if it is single sku in which case it** **says single sku iblpn or if it is multi sku then it will say multisku iblpn.**	Dynamic	None
facility_shipfrom_name_address **(qs, args=u")**	**Facility ShipTo Fields - generates an address in the following format: facility_name facility_address facility_address2 facility_city, facility_state facility_zip facility_country**	Dynamic	None
facility_time (qs=None, args=u'd/%m/%Y - %H:%M')	**Facility Time - the current time and date when the document or label is printedbased on the facility time zone location.**	Dynamic	**Date Format String.** ***http://strftime.org/*** **Default: "%d/%m/%Y - %H:%M"**
get_container_num_for_item_single_sku_on_order (qs, args=u")	**Get Container Number For Item Single Sku On Order - This is only for labels or documents that are container based. This would not**	Dynamic	None

Function	Definition	Data Type	Argument
	make sense for an order label for instance. Checks to make sure the status is Packed or Higher but not cancelled. Checks to make sure the Allocation is in Completed status. This function gets what container number out of the total number of containers on an order with regards to an item.		
get_country_code (code=u'US')	Get Country Code Pass in any form of United States or Canada and get back the 2 letter country code.	Dynamic	code Default: 'US' Only Use With Expressions
get_country_name (code=u'US')	Get Country Name Pass in 2 letter code, 3 letter code, or dialing code of any country and get back the country Name.	Dynamic	code Default: 'US' Only Use With Expressions
get_display_string (qs=None, args=u'')	Get Display String Gets the information on the object being processed. For instance if this was an oblpn label then this would get the information on the ob container.	Dynamic	None
get_ib_shipment_ids_from_vendor_code (vendor_code=u'')	Get In Bound Shipment IDs From Vendor Code Gets all of the in bound shipment ids for a vendor code.	Dynamic	: vendor_code Default: '' Only Use With Expressions
get_julian_day (date_str, args=u'%Y-%m-%d')	Get Julian Day The Julian Day Server date when the document or label is printed.	Dynamic	Date Format String. *http://strftime.org/* Default: "%d/%m/%Y - %H:%M

Function	Definition	Data Type	Argument
get_maxi_class_of_service (indicator)	**Get Maxi Class Of Service - This is used for maxi class service lookup for UPS Maxi Code Generation.**	Dynamic	**indicator Default: None** **Only Use With** **Expressions**
get_maxicode (qs, args=u")	**Get Maxi Code - ********** This is still in Development ********** ---------- UPS Maxi Code Generation.**	Dynamic	None
get_num_containers_for_item_single_sku_on_order (qs, args=u")	**Get Container Number For Item Single Sku On Order - This is only for labels or documents that are container based. This would not make sense for an order label for instance. Checks to make sure the status is Packed or Higher but not** **cancelled. Checks to make sure the Allocation is in Completed status. This function gets the total number of containers on an order with regards to an item.**	Dynamic	None
get_pre_pack_location (qs, args=u")	**Get Pre Pack Location Gets the location off of the allocation.**	Dynamic	None
get_pre_pack_part_range (qs, args=u")	**Get Pre Pack Part Range - Based on the values in the PRE_PACK_VARIABLE_PARTS** **company parameter, these are excluded from the parts seperated by a - that shows all of the child parts that make up this prepack.**	Dynamic	None
get_pre_pack_part_range_fields (qs, args=u")	**Get Pre Pack Part Range Fields - Based on the values in the PRE_PACK_VARIABLE_PARTS**	Dynamic	None

Function	Definition	Data Type	Argument
	company parameter values refer to the fields which are excluded from the rest of the fields in the prepack. The remaining fields that make up a prepack is returned.		
get_pre_pack_ratio_range (qs, args=u")	Get Pre Pack Ratio Range - Based on the values in the PRE_PACK_VARIABLE_PARTS company parameter these are excluded from the parts seperated by a - that shows all of the child units relating to the parts that make up this prepack.	Dynamic	None
get_store_nbr (qs, args=u")	Order Get Store Number Checks Destination Facility Code Then Checks Shipto Facility Code Then Checks external route Then Check Customer Number	Dynamic	None
get_total_cases_ob_containers (qs, args=u")	Get Total Cases Out Bound Containers Get the total number of cases from the Out Bound Container.	Dynamic	None
get_total_num_containers (qs, args=u")	Get Total Number Containers Looks at the allocation from inventory container pallet and for all containers that are less than shipped status it counts them and returns the number of containers.	Dynamic	None
get_total_num_containers_on_load (qs, args=u")	Get Total Number Containers On Load - Checks to make sure the status is Out Bound Ready, In Packing, Packed, Loaded, or Shipped. This function gets the total number of containers on the load.	Dynamic	None
get_total_num_out_bound_ready_containers_on_order (qs, args=u")	Get Total Number Out Bound Ready Containers On Order Checks to make sure the status is Out Bound Ready or Higher but not cancelled and not in Picked,	Dynamic	None

Function	Definition	Data Type	Argument
	or Picking. This function gets the total number of containers on the order.		
get_total_num_packed_ containers_on_order (qs, args=u")	**Get Total Number Packed Containers On Order Checks to make sure the status is Packed or Higher but not cancelled. This function gets the total number of containers on the order.**	Dynamic	None
get_total_num_packed_items_ on_order (qs, args=u")	**Get Total Number Packed Items On Order Checks to make sure the status is Packed or Higher but not cancelled. This function gets the total number of items on the order.**	Dynamic	None
get_total_packs_ob_containers (qs, args=u")	**Get Total Packs Out Bound Containers Get the total number of packs from the Out Bound Container.**	Dynamic	None
get_total_qty_for_lpn_with_ prepacks (qs, args=u")	**Get Total Quantity For Lpn With Prepacks This function gets the total number of child units for a container**	Dynamic	None
get_total_qty_ob_containers (qs, args=u")	**Get Total Quantity Out Bound Containers Get the total number of units from the Out Bound Container.**	Dynamic	None
get_total_weight_packed_ containers_on_order (qs, args=u")	**Get Total Weight Packed Containers On Order Checks to make sure the status is Packed or Higher but not cancelled. This function gets the total Weight of the containers on the order.**	Dynamic	None
get_ups_routing_code (zip_code, country_code=u'US')	**Get UPS Routing Code This is used for routing code and version for UPS Maxi Code Generation.**	Dynamic	**zip_code, country_code Default: zip_code = None, country_code = 'US' Only Use With Expressions**

Function	Definition	Data Type	Argument
get_us_state_code (code=u")	**Get State Code Pass in state name and get back the 2 letter state code.**	Dynamic	**code Default:** **Only Use With** **Expressions**
get_us_state_name (code=u")	**Get State Name Pass in 2 letter state code and get back the state name.**	Dynamic	**Arguments:** **code Default:** **Only Use With** **Expressions**
get_x_of_y_for_container_on_load (qs, args=u")	**Get X Of Y For Container On Load This is only for labels or documents that are container based. This would not make sense for an order label for instance. Checks to make sure the status is Out Bound Ready, In Packing, Packed, Loaded, or Shipped. This function gets what container number out of the total** **number of containers on an order.**	Dynamic	None
get_x_of_y_in_out_bound_ready_containers_on_order (qs, args=u")	**Get X Of Y In Out Bound Ready Containers On Order This is only for labels or documents that are container based. This would not make sense for an order label for instance. Checks to make sure the status is Out Bound Ready or Higher but not cancelled and not in Picked, or Picking. This function gets what container number out of the total number of containers on an order.**	Dynamic	None
get_x_of_y_in_packed_containers_on_order (qs, args=u")	Get X Of Y In Packed Containers On Order	Dynamic	None

Function	Definition	Data Type	Argument
	This is only for labels or documents that are container based. This would not make sense for an order label for instance.Checks to make sure the status is Packed or Higher but not cancelled. This function gets what container number out of the total number of containers on an order.		
is_carrier_fedex(qs=None, args=u")	**Is Carrier FedEx Displays True if the carrier used was FedEx if not then it returns False.**	Dynamic	None
is_carrier_ups (qs, args=u")	**Is Carrier UPS Displays True if the carrier used was UPS if not then it returns False.**	Dynamic	None
is_shipvia_fedex_express (qs, args=u")	**Is ShipVia FedEx Express Displays True if the ship via service level used was FedEx express. If not then it returns False.**	Dynamic	None
is_shipvia_fedex_ground (qs, args=u")	**Is ShipVia FedEx Ground Displays True if the ship via service level used was FedEx ground. If not then it returns False.**	Dynamic	None
is_shipvia_fedex_smartpost (qs, args=u")	**Is ShipVia FedEx Smart Post Displays True if the ship via service level used was FedEx Smart Post. If not then it returns False.**	Dynamic	None
order_header_cust_name_address (qs, args =u")	Order Header Customer Address Fields generates an address in the following format: customer_name customer_address customer_address2	Dynamic	None

Function	Definition	Data Type	Argument
	customer_city, customer_state customer_zip customer_country		
order_header_shipto_name_address (qs, args =u'')	Order Header ShipTo Address Fields generates an address in the following format: shipto_name shipto_address shipto_address2 shipto_city, shipto_state shipto_zip shipto_country	Dynamic	None
shipto_facility_name_address (qs, args =u'')	Order Header ShipTo Facility Address Fields generates an address in the following format: shipto_facility_name shipto_facility_address shipto_facility_address2 shipto_facility_city, shipto_facility_state shipto_facility_zip shipto_facility_country	Dynamic	None
wave_time (qs = None, args=u'%d/%m/%Y')	Wave Time -The wave time and date when the document or label is printed based on the facility time zone location.	Dynamic	Arguments: Date Format String. *http://strftime.org/* Default: "%d/%m/%Y - %H:%M"

Function	Definition	Data Type	Argument
get_lowest_location_by_location_pick_seq	Get Lowest Location By Location Pick Sequence -Returns the location string of the lowest location by location pick sequence.	Dynamic	None
get_lowest_pick_zone_by_location_pick_seq	Get Lowest Pick Zone By Location Pick Sequence- Returns the location string of the lowest pick zone by location pick sequence	Dynamic	None
get_total_ib_shipment_container_qty	Total Inbound Shipment or Inbound Containers - Returns the total of inbound containers quantity if a container is passed in. Returns the total of inbound shipment details quantity if an ib_shipment_id is passed in.	Dynamic	None
rename_oblpn_nbr_once	Rename OBLPN Number Once - This function renames the oblpn.	Dynamic	None

10 Text

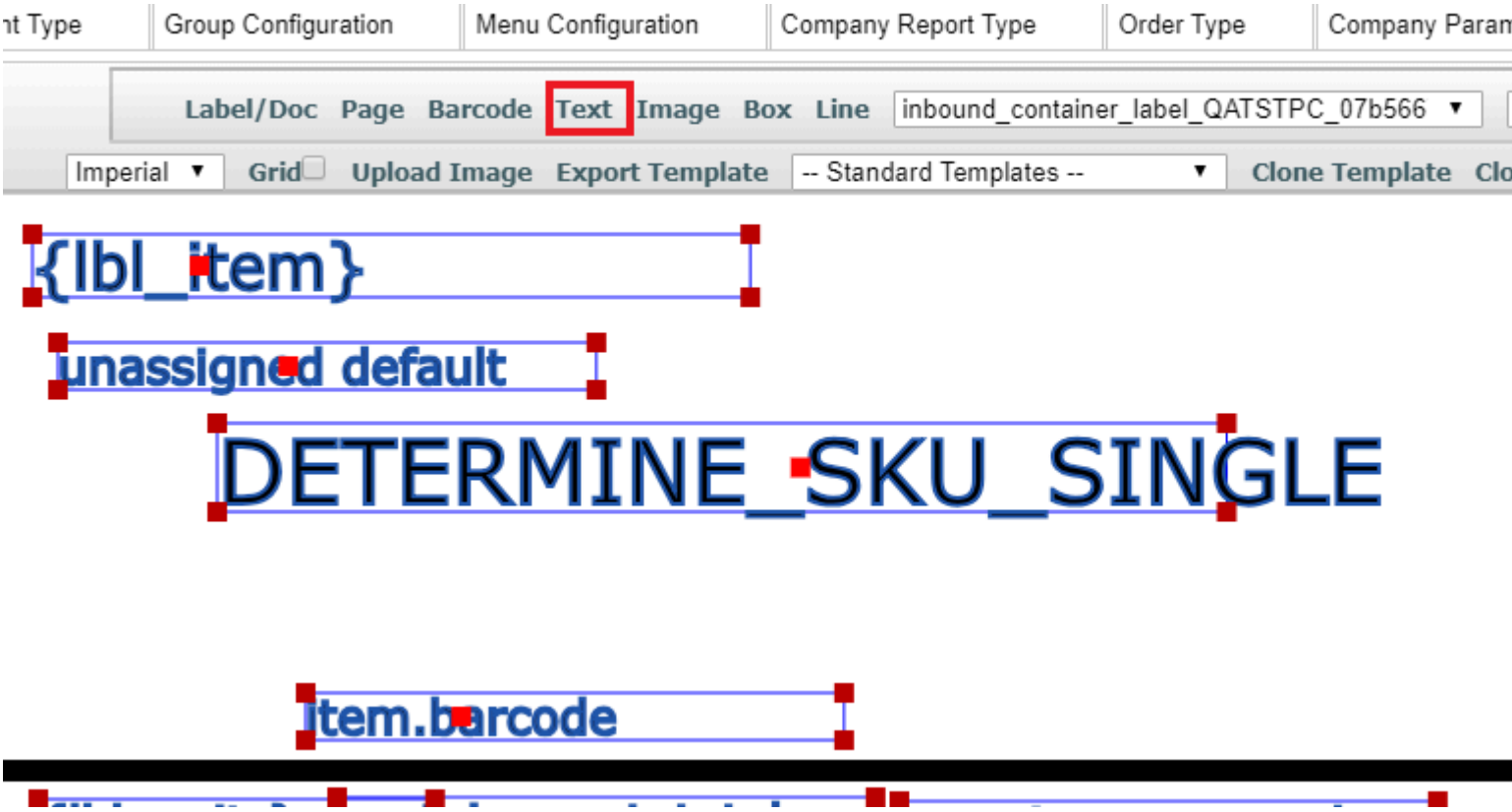
Text

Once you click in the label designer text, the **Edit Text** pane appears on the right and allows you to make edits.

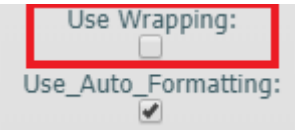
Change the values you want to change and then click **Save.**

> **Note:** Text should be limited to the number of characters in your length field, otherwise, text and words can run together during printing. Also, when you are designing your label, keep in mind the size of your paper.

> **Note:** Your changes will not be applied until you click **Save.**



*You have the option to de-select **Use Wrapping** if you want to truncate any of your text elements.*



Edit Text Definitions

Edit Text Field	***Description***
Creating Text	To create new Text, click the text at the top of screen in order.
Move Text	You can move text by clicking on it and then dragging the text to where you want it to be positioned. The text will automatically save its position when you are done dragging. Another way to move text is by entering its x and y coordinates. X = 0 and Y = 0 is the top left corner of the page. The bottom right corner are X = page width and Y = page height. So by entering the number you want for positioning it will allow you to put the text where you want it. After you type a number and then hit tab or click out of the edit box the text will move to that location. In order to save your change you have to click the save button.
Data Type	Allows you to set your data to 1 of 3 values. Static means your text will never change like a title but this still allows for literals to be used for different languages. Only the Value field can be used with Static Data Types. Dynamic means you will only have one value that changes depending on data. Example would be an order header order number on an order. Dynamic Repeating is a means you can get back more than one result typically used in a table of values. For example, order detail where the order quantity is displayed for an order. You can get back more than one order detail for a single order. Column Name, Expressions, and Functions can be used with Dynamic Data Type. Column Name, and Functions can be used with Dynamic Repeating Data Type.
Value	Allows for a static unchanging text good for things like the title of a column etc.
Page Numbers	Allows you show the current page or the total number of the pages depending on which one you select.
Font Type	Allows you to select different fonts for your pdf or label for that particular text. Users can not add fonts because these documents are generated on the server. So the fonts must exist on the server when they are being flattened for pdfs etc. Fonts for Zebra must exist on the Zebra printer that is being printed on. Pdf fonts are close approximates open source versions of proprietary fonts.
Length	How many characters long on a single line before text wrapping or cutting off the text at this value based on use wrapping option.
Height	How many lines will fit on a single page.
Spacing	How much spacing between lines of text.
Use Wrapping	If turned on, wraps text to the next line based on spaces between words. If turned on then your height should be set to something greater than 1 unless you want to cut off by word and not character length. If turned off then the text gets cut off at that exact character length.

Edit Text Field	***Description***
Use Auto Formatting	If turned on, auto removes \n new character lines the same as if the enter key was hit. This is ideal for data coming in from a column name mapping. For a custom function or standard function typically it is best to have this turned off because they may have special formatting.
Table Name	Group Dynamic Repeating text elements together as a table to be displayed on the page.
Display Once Per Element	This is for only Dynamic Repeating Data Types. Displays the data one time for each element that gets returned from the database. So say the data has more than one line and say we are displaying multiple order details for an order for multiple orders. We only show the order number one time for each group of order details corresponding to that order.
Bold	Makes all of the text bold in this text element.
Italicize	Makes all of the text italicize in this text element.
X	Position the Text on the X-Axis based on the number of dots of ink or thermal burning on a page.
Y	Position the Text on the Y-Axis based on the number of dots of ink or thermal burning on a page.
Size	How large the text is based on the font point system.
Rotate	Rotates the text. For Labels you can rotate by 0, 90, 180, 270 degrees as valid rotations. For Documents PDFs you can rotate between 0 - 90 degrees.
None Mapping	When there is an empty value also known as None what do you want to display? Default display value is - Note this is not the same thing as 0 this means no value at all.
Page	Set what page this text will appear on.
Disabling Buttons While Loading	The buttons at the top making a new page, document, text, barcode, box, or line, etc. Are grayed out and disabled while it is loading.
Multiple Page Templates	Your Dynamic Repeating text type needs to be the same height across all of the different text elements in your table on the page. Table Name allows you to group these dynamic repeating text elements together in a single table. When you do your mapping of your text element on the different pages you do not have to do anything to make them match up to the other text elements on the other template pages. This matching of the text elements on the different template pages happens automatically when you print your template. Some smart code figures out how the data should get printed across the pages, to what fields, for how many pages, and for what table groupings.
AIT Database Query To Help You Find The Expressions In A Document Or A Label	Select **expression, dynamic_single** from **doc_label_text.** where page_id in (select id from doc_label_page where doc_label_id = (select id from doc_label where code = 'shipping_label_CRAFTSY_f3e95bd'))
Supported Special Characters	¡ ¢ £ ¤ ¥ ¦ § ¨ © ª « ¬ ® ¯ ° ± ² ³ ´ ▯ ¶ · ¸ ¹ º » ¼ ½ ¾ ¿ À Á Â Ã Ä Å Æ Ç È É Ê Ë Ì Í Î

Edit Text Field	*Description*
	Ï Ð Ñ Ò Ó Ô Õ Ö × Ø Ù Ú Û Ü Ý Þ ß à á â ã ä å æ ç è é ê ë ì í î ï ð ñ ò ó ô õ ö ÷ ø ù ú û ü ý þ ÿ

11 Uploading and Editing Images

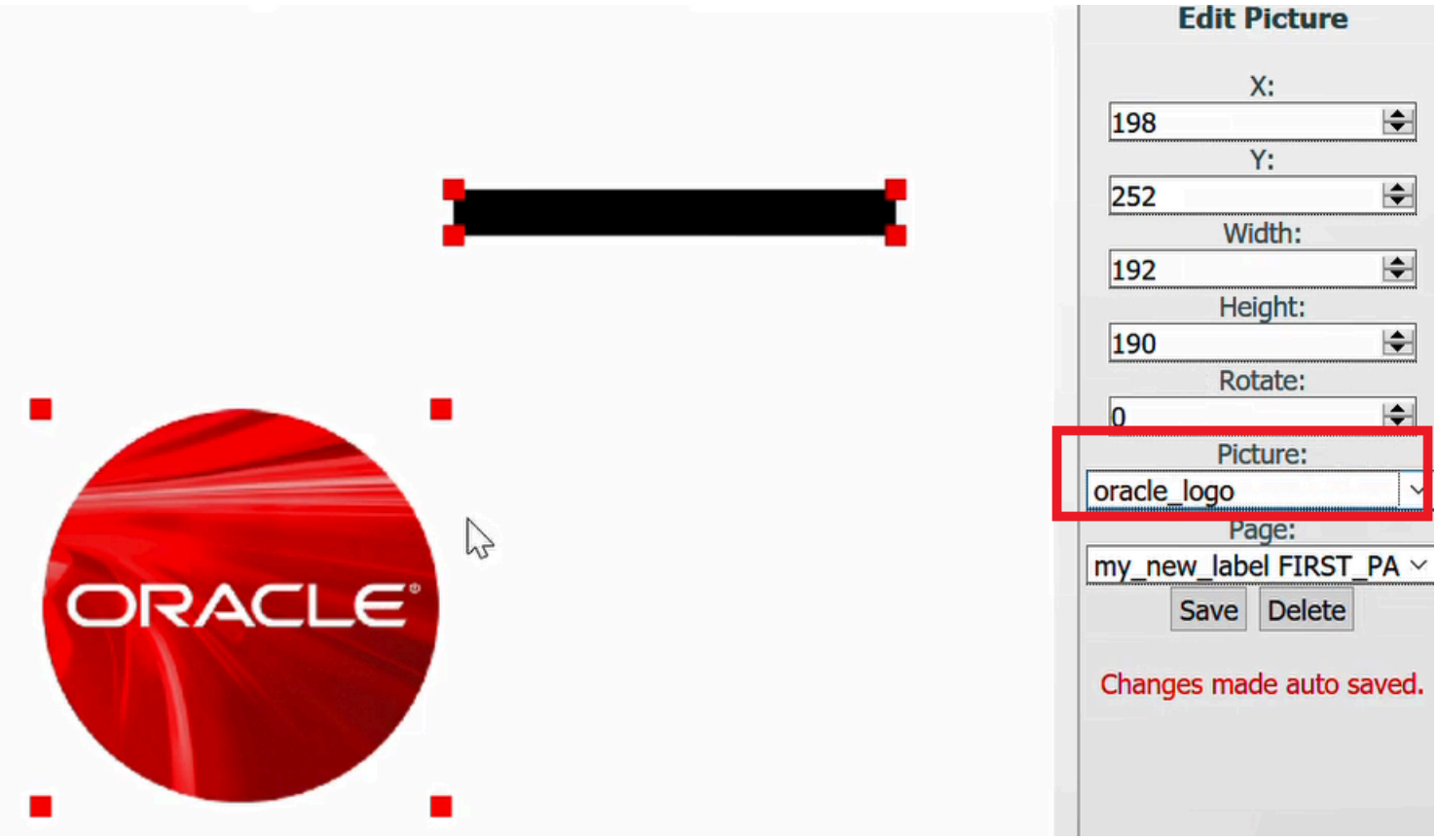
Uploading and Editing Images

When you click on an image, the **Edit Picture** pane appears on the right and allows you to edit images.

*From the **Label Designer** tab, click **Image** and then **Upload Image**:*

Once your image is uploaded, a message appears letting you know that your upload was successful.

*You can then select your image from the **Picture** drop-down:*



Make any edits to the image position (X and Y drop-downs), image width and height (Width and Height drop-downs), and page that you like.

*Click **Save**.*

Page

Set what page this image will appear on.

Disabling Buttons While Loading:

The buttons at the top making a new page, document, text, barcode, box, or line, etc. Are grayed out and disabled while it is loading.

Working with File Formats

When you are naming a file for an image only use the following characters:

- a-z
- A-Z
- 0-9

Anything else should not be used in a file name. You also should not start the name of a file with a number.

Avoid exporting images with Adobe products unless you are well versed in Graphic Design. These images will not load in our software. Please install GIMP via the following url: *https://ninite.com/*.

If the image is failing to upload first make sure it is named correctly. If it is still failing, there is a good chance the image is in adobe proprietary format only meant to work in adobe software.

To resolve issues with images failing to upload, complete the following steps:

1. Open GIMP and select **File**->**Open.**
2. Select your image to open it.
3. Select **File**->**Export** also known as **File**->**Overwrite Your Image File Name.**
4. Set the quality to 100% and click **Export**. This should fix the file issue.

12 Printer and Page Setting Guidelines

Printer and Page Setting Guidelines

- You should leave a margin of 0.25 inches on the sides and top and bottom of a page due to physical printer limitations.
- Most printers default to a mode that can only print on a portion of 8.5 x 11 paper. Your printer for example, may only be able to "address" an 8.0 x 10.7 inch portion of the paper.
- The print head must have enough time to accelerate for example and get up to a constant speed before spraying ink and must decelerate at the opposite side of the page, creating the left/right borders. The paper itself must be able to load and be moved accurately by the rollers, which creates the top/bottom borders. These limitations mean that there is a "printable area" on the page that is smaller than the paper itself, and that this printable area is an area inside which the printer can operate optimally to produce the highest quality prints. There are often driver options that can affect these limitations.

Labels in Additional Languages

You may need to install additional fonts for languages other than English. See *Fonts for Zebra Printers* for information about additional fonts. You can also contact *Zebra support*.

Note for Arabic Labels: In order to print Arabic Labels, you need to install the Swiss 721 font into your Zebra printer in one of the following memory locations:

- A, B, or E.

E is probably the best memory location to install this font into, but you should review your printer to determine the best memory location.

Refer to *this Zebra link* for the Swiss 721 font.

You will need to select the Arabic font in the text element where you want the arabic text and select the Arabic font for the memory location you installed the font in your printer.

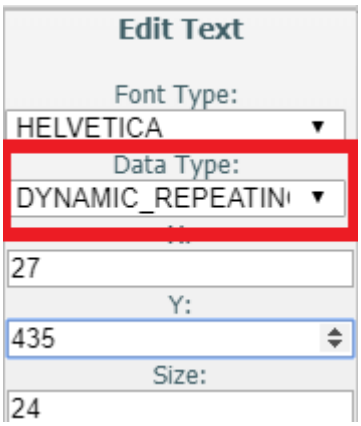
Note: If you need additional support installing fonts, please contact *Zebra support*.

Multi-Page Printing

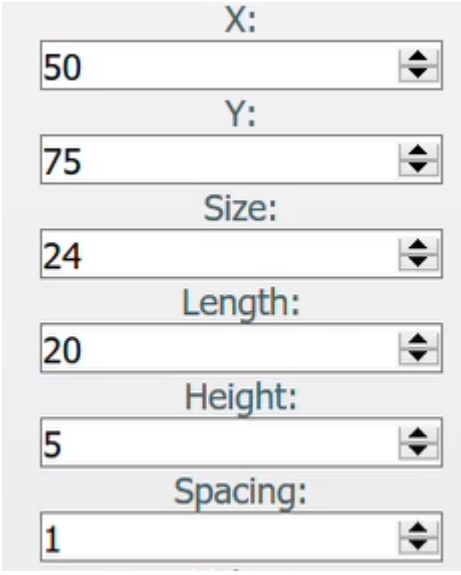
Multi-Page printing allows you to print across multiple labels, documents, or .pdf files. To set up Multi-Page printing for your label or document, you need to set up dynamic repeating fields that are linked together in a table of values. Height needs to match item table.

Note: Label Designer currently only supports static page layout. It does not support dynamic page layout.

1. From the **User Defined Templates** drop-down, select the label you would like to edit.
2. Click **Text** to edit text settings. You can click on each element in your page to specify and save different settings if you like.
3. For the **Data Type**, select **Dynamic_Repeating**.

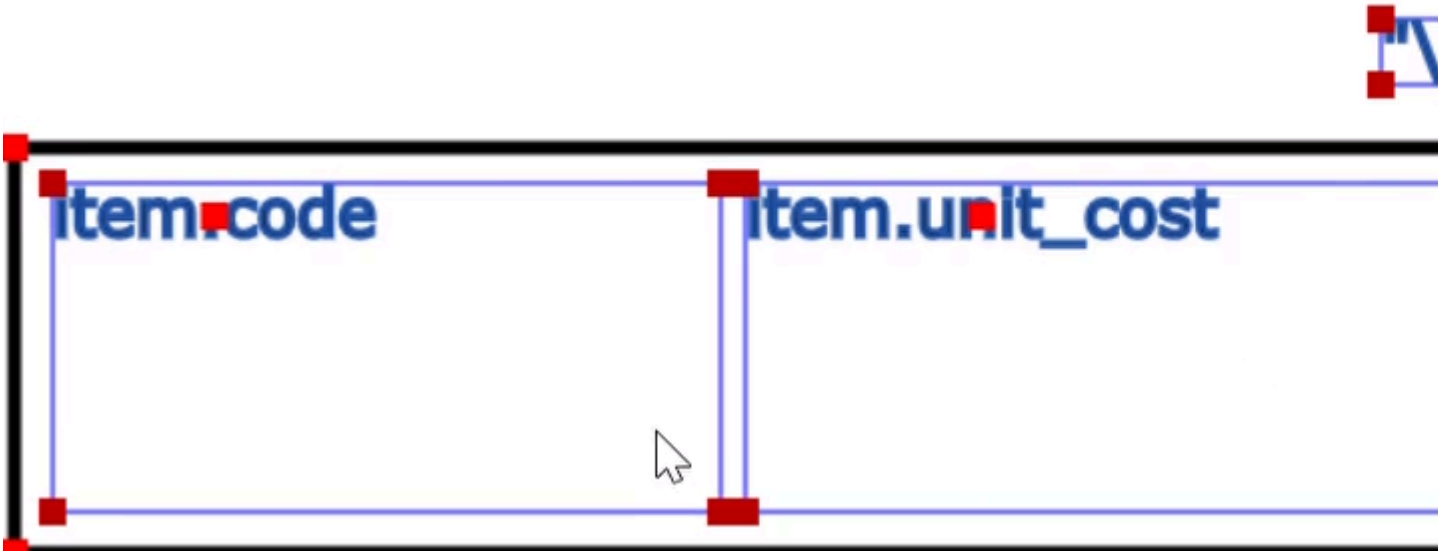


4. Next, specify your settings **for X, Y, size, length, height**, and **spacing**. It is not recommended to set the height for dynamic repeating fields as 1 as this will create unneccesary wasted paper during printing.



Note: It is important that all length and height settings for columns that are next to one another in a table line up for precise formatting and printing. For every multi-template page you have, you have to have the same field settings for each table such as height, spacing, column name, and table name to prevent errors.

5. Specify a column name.

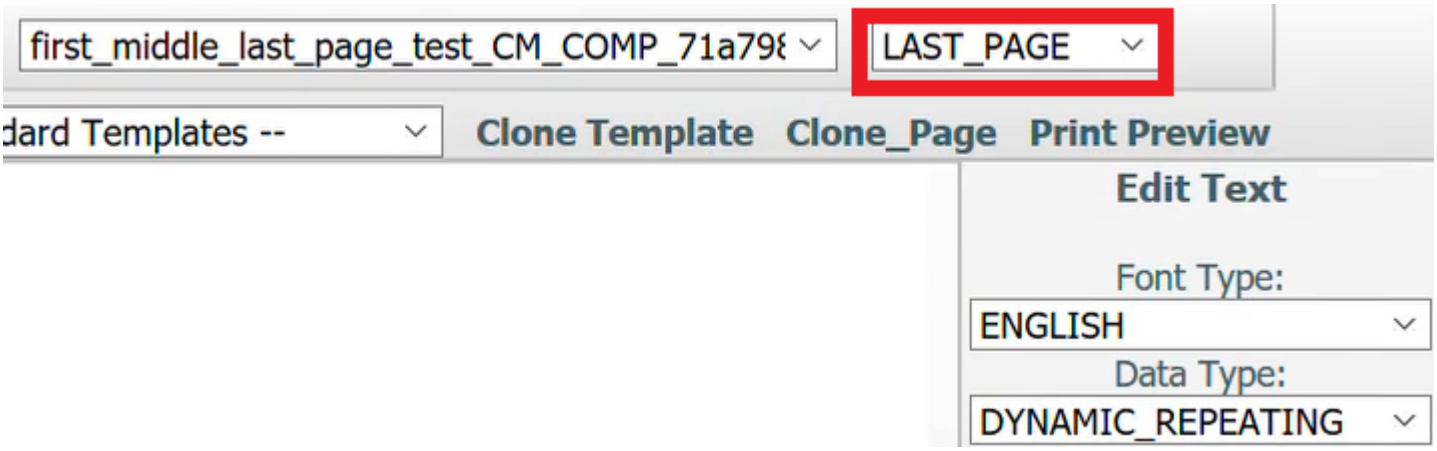


6. Enter the table name you want your page fields associated with.

   The system will automatically associate the fields with the table. Also, once field specifications are aligned, they will associate with the same fields on middle and last pages until it moves to the next template.

   > **Note:** It is important that for the first page, the settings (for example, height/length/spacing) align with the table settings. If they do not match, you will receive an error.

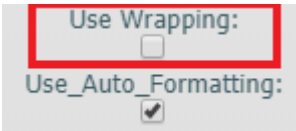
7. From the **Pages** drop-down, update your settings for your first, middle, and last page templates.



   Keep in mind that text may need to wrap within columns, which can affect how many lines are printed in each column.

8. Select **Use Wrapping** and **Use_Auto_Formatting** for Dynamic Repeating pages.

   Use Wrapping keeps the text wrapping consistent throughout your pages. Use Auto Formatting automatically cleanses the data so that your formatting remains consistent for printing multiple lines and pages.



The system will automatically associate the fields with the table. Also, once field specifications are aligned, they will associate with the same fields on middle and last pages until it moves to the next template.

> **Note:** It is important that for the first page, the settings (for example, height/length/spacing) align with the table settings. If they do not match, you will receive an error.

Print Multiple Serial Numbers

You can print multiple serial numbers using our new Label Designer Function, SERIAL_NBRS_FROM_OBLPN. This function allows multiple serial numbers for a SKU to be scanned in a single scan.

This new function is available for OBLPN Labels, OBLPN Content Labels, and Shipping Label, as well as all other labels. It works best in two-dimensional barcode labels such as QR Barcode Type, as it will embed a large number of characters allowing you to provide more information.

> **Note:** You can also use this function on Multiple SKU items, but the system will only return serial numbers and not include the related SKU information.

Enable Printing Multiple Serial Numbers

To enable printing multiple serial numbers in Label Designer, set up the following function details:

- Data Type: Dynamic
- Function: SERIAL_NBRS_FROM_OBLPN
- Function Arguments example: {delimiter:",",max_chars:0}

Where,

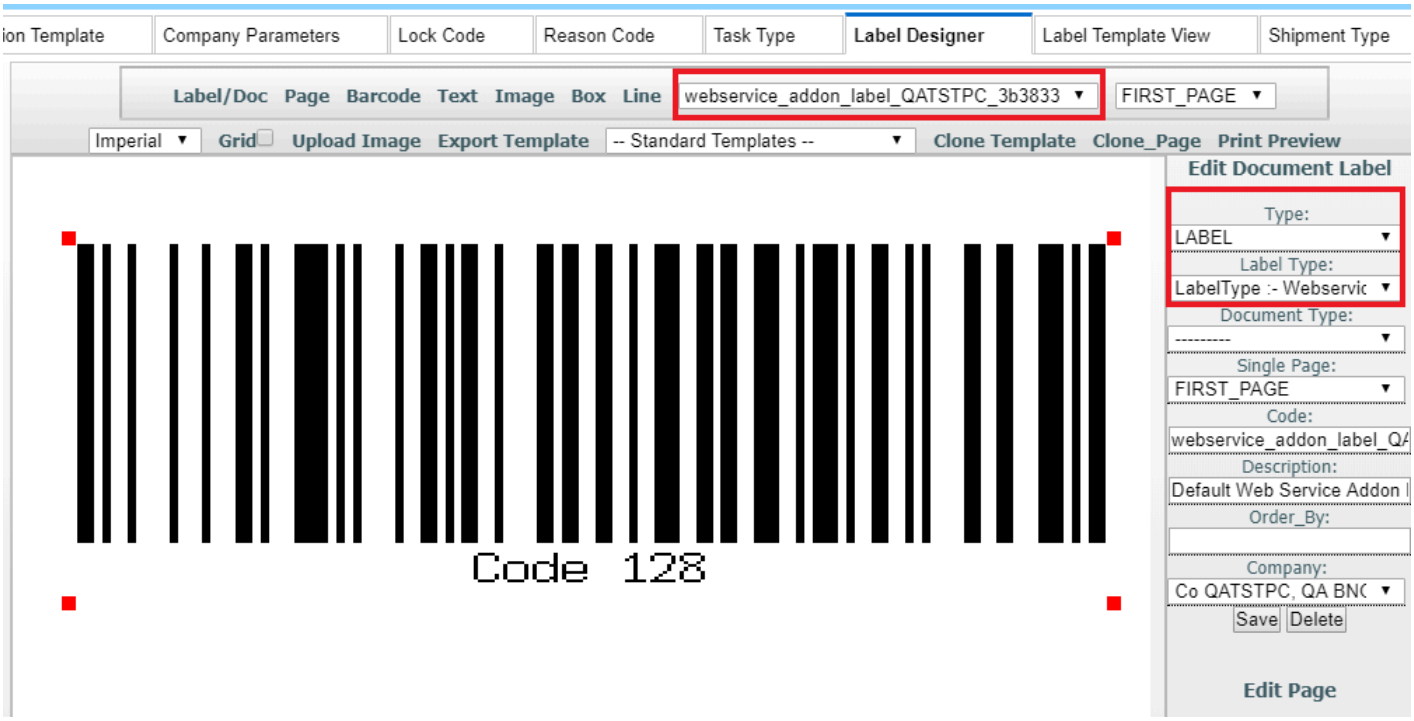
- the delimeter could be comma (,), pipe (|), colon (:)
- max chars: it's the maximum number of characters you want to display in the label. 0 means unlimited or instead of the function you could use
- Expression : serial_nbrs_from_oblpn(get_qs(), args="{\"delimiter\":\",\", \"max_chars\":0}")

13 Label Template Setup

Label Template Setup

You will setup your Label Templates via the **Label Designer** tab. To create the Label Template, complete the following steps:

1. Select **webservice_addon_label** from the User Defined Templates drop-down.
2. Select **Label** from the Type drop-down.
3. Select **Webservice Label Type** from the Label Type drop-down.



Base Label Templates

The following table lists the available base label templates for the various label types in Oracle WMS Cloud:

Label Type	Label Names
Web Service labels for shipping	ws
Shipping	shipping.lbl
Price	pricelabel.lbl

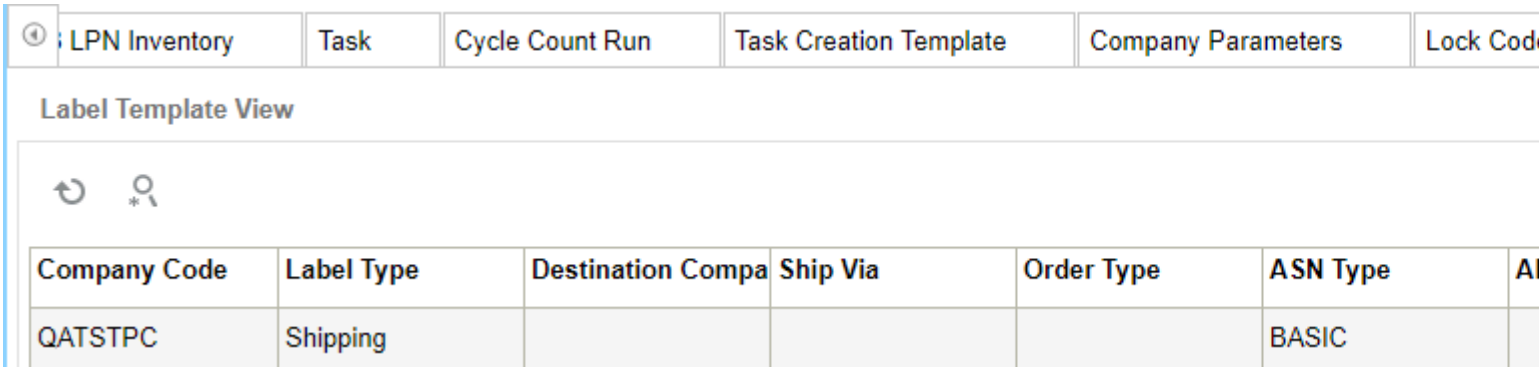
Label Type	Label Names
Item	item.lbl
Task	task.lbl
Blind	blindcontainer.lbl
IB Container	ibcontainer.lbl
Location	location.lbl
Order	order.lbl
OB Container	obcontainer.lbl
IB Shipment	ibshipment.lbl
Serial number	serialnbr.lbl
Work order	work_order.lbl

Label Template View

You can further refine your label settings from the **Label Template View** tab. For a webservice label, you will want the following settings:

1. **Destination Company** should be blank.
2. Select **Shipping** from the **Label Type** drop-down.
3. Enter ws for webservice label (all lowercase) for the **Template Name**.
4. Select webservice_addon_label from the **Label Designer** drop-down.

For shipping labels that are not webservice labels, type a description of the label in the template name and then select the label designer template you want to use.



Note for Shipping Labels:

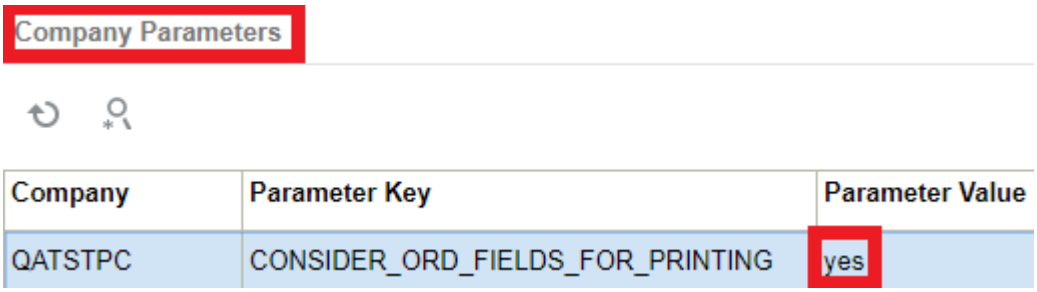
A default value of **yes** will consider order header ship via code as part of the ship via check which is required for some LTL TL flows. Setting this to **no** will only allow carrier LPN ship via code.

The **Label Template View** tab is where you configure label templates to print for different buttons.

1. To configure a label template to print with certain specifications from the **Label Template View** tab, click **Create**.
2. Complete the fields for your label.

   The doc label code you created for your template will be available from the **Template Name** drop-down.

IMPORTANT: The company parameter CONSIDER_ORD_FIELDS_FOR_PRINTING has to be set to Yes for label template setup to work for OBLPNs.



Note: the **No of Labels per Row** field is used for the number of labels per row for price labels only. For price labels, this field allows you to print a label that has more than one label across.

The following table details the corresponding screens where you can go to print out various label types:

Label Type	Associated Screen to Print Label
• Blind Container	• IBLPN
• IB Container	• Inbound Shipments (Print lpn Labels option), IBLPN
• Item	• Items
• Location	• Locations
• OB Container	• OBLPNs, Print {oblpn} Label
• Order	• Order Headers, Order
• Task	• Pallets
• Shipping	• OBLPNs, Print Shipping Label
• IB Shipment Label	• Inbound Shipments
• Serial Number	• IB Shipment Serial Number, Serial Number Inventory
• Webservice Addon Label	• OBLPN (Print Shipping Label because this is an Addon to Shipping Label)
• OBLPN Contents	• OBLPNs, Print OBLPN Contents
• Work Order	• Work Order
• Price Labels	• Inbound Shipments, Inbound Shipment Detail