

Functional Tips for ScanWorkX



This month's tip for getting more from your ScanWorkX implementation!

ScanWorkX for D365 – Application Field Editor Overview

-By Harrison Brazelton

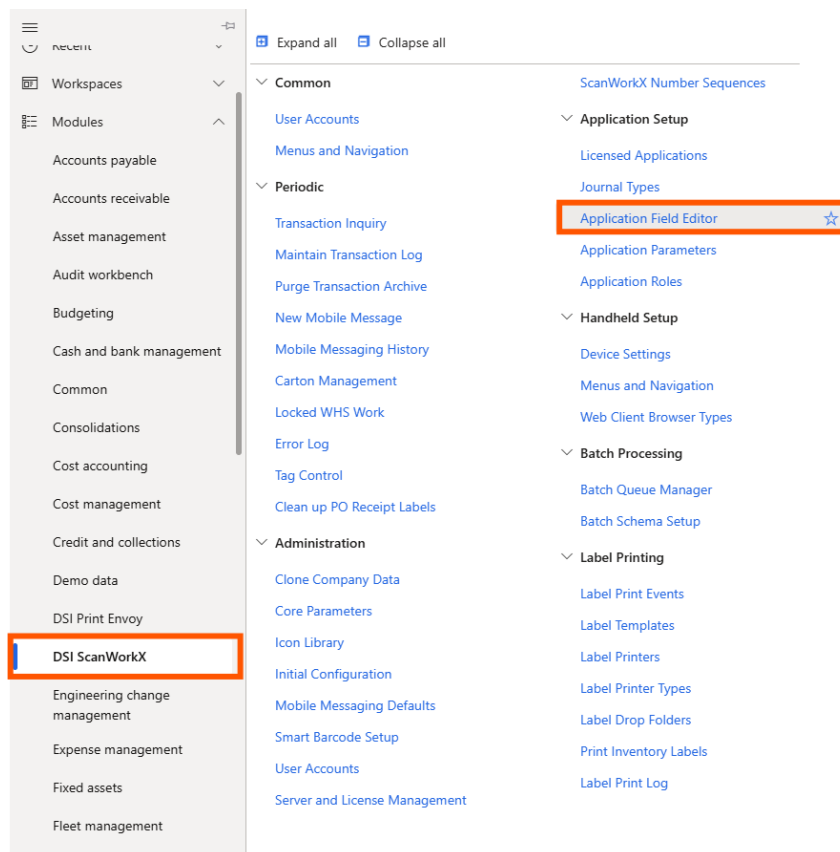
harrison.brazelton@nextw.com

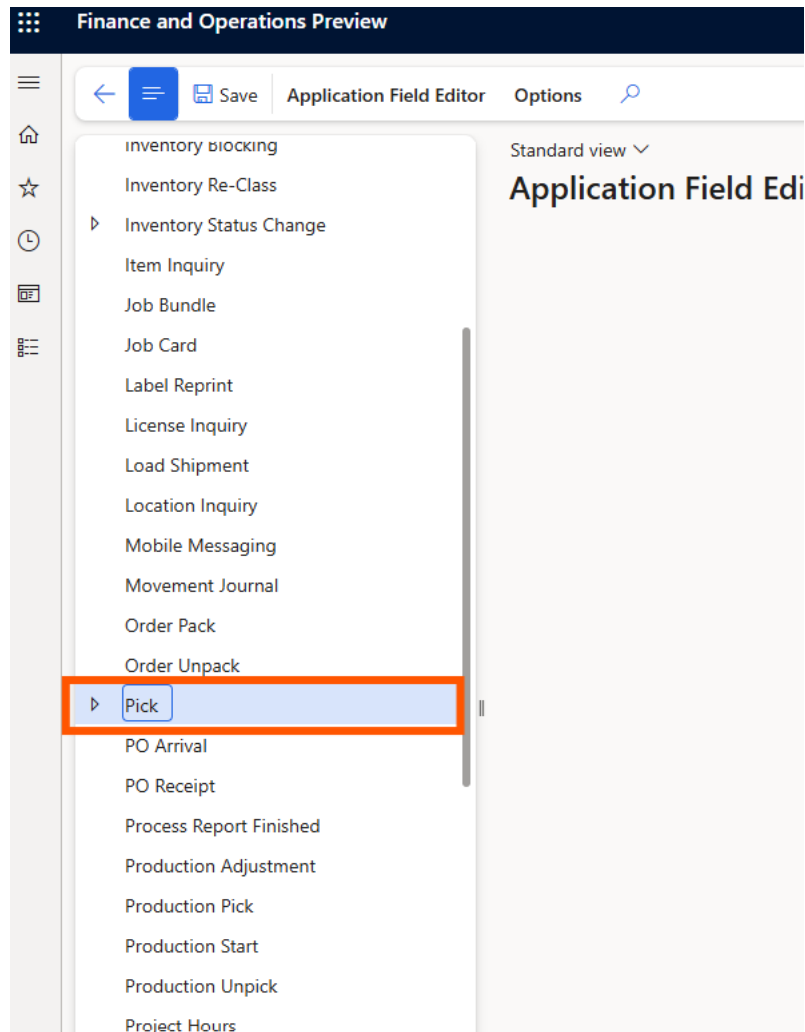
In this month's edition of the ScanWorkX newsletter we will be discussing the 'Application Field Editor' and how you can leverage this form to customize your ScanWorkX experience. The Application Field Editor allows you to make customizations on the application's fields themselves, coupled with cloned applications and the Application Parameters form, this allows for an abundance of customization options for each individual application.

applications and the Application Parameters form, this allows for an abundance of customization options for each individual application.

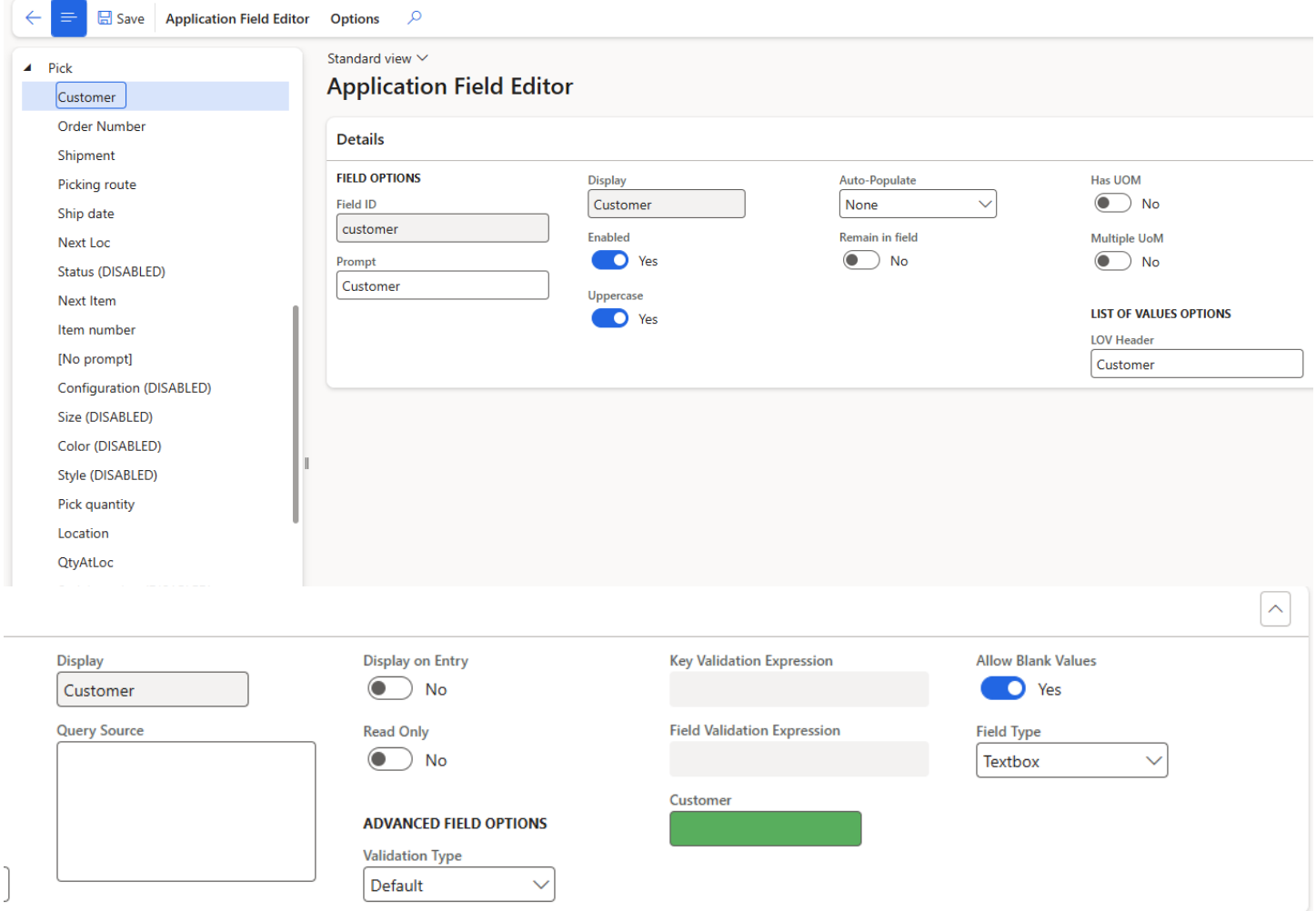
Application Field Editor

Navigate to the following path: DSI ScanWorkX – Application Setup – Application Field Editor.





Once you have opened the form you will see all the ScanWorkX applications and cloned applications listed here, you will need to select one of them to expand the menu and display all the available fields for the application. In this example, we will be expanding the Pick application.



Now that I have expanded the menu, I have gone ahead selected the 'Customer' field. This brings up the Application Field Editor screen that allows us to change how the field looks and behaves. We will now go over each of the options pictured in the two screenshots above.

Field ID – This represents the name of the variable used in the ScanWorkX code.

Prompt – This field defaults to the label assigned to the field in D365. If the default value is not inherently meaningful to your company, this can be changed to any text.

Display – The purpose of this field is to display the label, if one is used, into readable text.

Enabled – This dictates whether the field can be used or not, if disabled you will not see the field nor will you be able to populate the value.

Uppercase - When enabled, the value of the field after being validated or auto-populated will be converted to all uppercase. This is useful for users as they can enter uppercase or lowercase values, e.g. a location called '123A' exists, the user can type '123a' or '123A' and the value will pass validation either way.

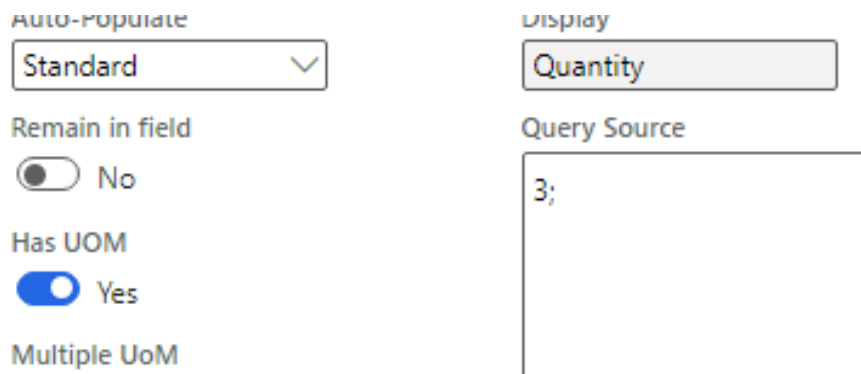
Auto-Populate – Attempts to automatically fill in an appropriate form value based on the LOV or in some cases the 'Query Source' field if it is populated. There are multiple settings available for this field:

None – The field will not attempt to automatically complete itself; the user must enter a value

Standard – The field will attempt to automatically populate (by running a LOV behind the scenes) if there is only one logical value available, as returned by the LOV function. * You can also use auto-populate with a single-value query source to set a static value for a field.

First/Last Value – If the LOV returns multiple values, you can elect to auto-populate the first or last returned value from the LOV, regardless of the number of results returned.

Example, the following settings will create default a value of 3.



The image shows a configuration interface for a field. On the left, there are several settings: 'AUTO-POPULATE' is set to 'Standard' in a dropdown menu; 'Remain in field' is a toggle switch set to 'No'; 'Has UOM' is a toggle switch set to 'Yes'; and 'Multiple UoM' is a label. On the right, there are two input fields: 'Display' is set to 'Quantity' and 'Query Source' contains the text '3;'. The 'Query Source' field is a text area with a border.

Note: If auto-populate results in a value that does not pass validation, the user will be returned to the last successfully entered field, but the error for the auto-populated field will be displayed. To assist the user, in these cases, the field name and value are displayed at the end of the error message to assist in determining what caused the error.

Note: If an application already defaults on a value into a field, the auto-populate feature cannot override that. You may need to request application modification.

Remain in field – Used when auto-populating a field, if enabled, the cursor will remain in the field after the initial auto-population, the user will have to validate again by hitting enter or scan a new value. If left disabled, the cursor will jump to the next field after validation as normal.

Has UOM – Determines if a unit of measure field is associated with the field (used for quantity values).

Multiple UOM – Some applications support multiple units of measure and can be toggled on and off for quantity fields. If enabled, the user will be able to choose any available unit of measurement in the unit of measurement LOV.

LOV Header – The string displayed in the list of value’s pop-up screen. If the ‘Display’ field is using a specific label, this value will be used. The following ‘Display’ field translates a label into readable text if one is used here.

Query Source – This allows List of Values to be customized. Leave this field blank to use the application’s default choices. Enter a semicolon-delimited list of choices to create static lists, with commas delimiting multi-line choices (e.g. “Blue,My Favorite;Red,Your Favorite”). If multi-line choices are used, every choice must have the same number of lines. Or enter the name of a Dynamics 365 for Operations Query object to create your own lookups.

You may define your unique LOV query with one of two query formats. Both queries shown below accomplish the same thing, returning the Location, License Plate and Available inventory in the LOV.

- This query format begins with a dollar sign and is similar to AX 2009 and 2012 queries:
\$select wMSLocationId, LicensePlateId, AvailPhysical from %1 where %1.PhysicalInvent > 0 && %1.ItemId == '{ItemId}' && %1.InventLocationId == '{InventLocationId}';InventSum
This query format implicitly adds the legal entity (the User's company id) to the query.
- This query form begins with the pound sign and is like T-SQL:
#select wMSLocationId, LicensePlateId, AvailPhysical from InventSum where PhysicalInvent > 0 AND ItemId = @ItemId AND InventLocationID = @InventLocationID AND DataAreald = @Company
This query format requires including the legal entity (DataAreald) in the query.

The substitution variables may be any application field, such as ItemID, as used in the examples.

These fields are also available:

- Company (legal entity)
- InventSiteId
- InventLocationId (warehouse)
- Identity (the user's logon username)

Display on Entry – If enabled, this will automatically display the list of values screen when the user enters this field. Can be used as a time save if you know users will always use the LOV on the field.

Read Only – If enabled, the LOV for the field will not return a value when an option is selected, but it will still display available values. E.g., a user is picking a location to put an item away in, the LOV will display available locations, but the user must physically scan the location or enter the value through text to validate the field. This can be used to prevent users from selecting a location but then putting an item away in a different location.

Validation Type – Type of client-side validation to use when users attempt to submit data. Values are tested at two levels, at the keypress and on posting the field. The key filter ignores keypresses that will obviously violate the client-side field validation rule. For example, if Y and N are the only valid choices in the field, the key filter will prevent the user from entering any value other than those two letters. The field validation evaluates the field when the user presses Enter or a postamble is received from the barcode scanner. At that time, the entire field is checked to ensure the data entered matches the validation criteria. If a match is not made, the value is immediately rejected client-side, without a round trip to the server.

Default – Client-side validation is OFF

YesNo – Users may only enter Y or N

Integer – Positive or negative, whole number values (e.g. 1, 2, 3, 0, -1)

MathInteger – Positive or negative, whole number values and appropriate math expressions for the integrated quantity calculator (e.g. 1, -1, 10x20, 20/2, 0)

Decimal – Positive or negative, decimal or whole number values (e.g. 1, 2.1, -3, 0)

MathDecimal – Positive or negative, decimal or whole number values and appropriate math expressions for the integrated quantity calculator (e.g. 1, -1, 2.1, 5.5+10, 10x20, 20/2, 0)

RegEx – User-defined regular expressions that are evaluated by keystroke and at the conclusion of the field (user presses enter or receives a scan gun postamble). Selecting RegEx from the list will activate additional fields: Key Validation Expression, Field Validation Expression, and Field Validation Test. I will not be going into further detail on RegEx expressions as we have covered this prior on an earlier newsletter topic (FUNctional tips Issue 1 – RegEx for ScanWorkX).

Key Validation Expression – Regular expression used to evaluate each keypress. Keypresses not matching the regular express are ignored. For example, if you wanted to validate the field for positive integers only, you could use the expression [0-9]. Any key the user pressed, other than a number key would be ignored.

Field Validation Expression – Regular expression used to evaluate the entire field. This is tested on the client when the user presses the Enter key or transmits a postamble from a barcode scanner. To continue the previous example, in order to allow a field to accept positive integers only, you could use the expression `^[0-9]+$` to ensure only numbers were entered, and at least one number was specified.

Field Validation Test – This field is used to test out your expressions in real time. When entering a value that is accepted by the expressions you have defined, the box will appear green as it is pictured below. If the value is not allowed by your expressions, the box will appear blue.

Key Validation Expression

Field Validation Expression

Lic Plate

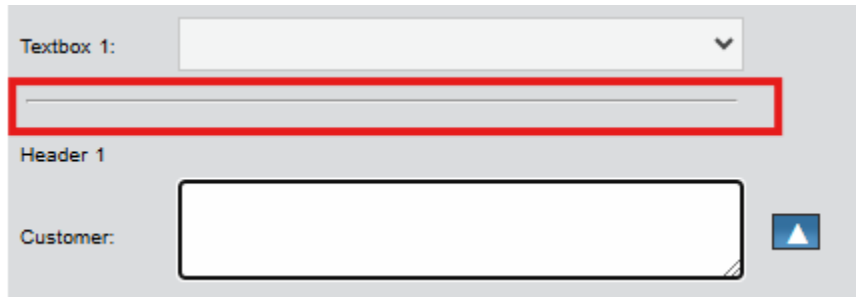
Allow Blank Values – When enabled, the user will be allowed to validate blank values, regardless of validation type.

Field Type – The format used to display and enter the field’s value on the ScanWorkX HTML client.

- Textbox – Default field type.
- Dropdown – Drop down box that displays choices based on the ‘Query Source’ field. This can be used in tandem with the Query Source to define a list of static values you want users to be able to choose from. E.g., on a ‘status’ field, you want the users to always be able to choose from ‘AVAILABLE’ or ‘UNAVAILABLE’.

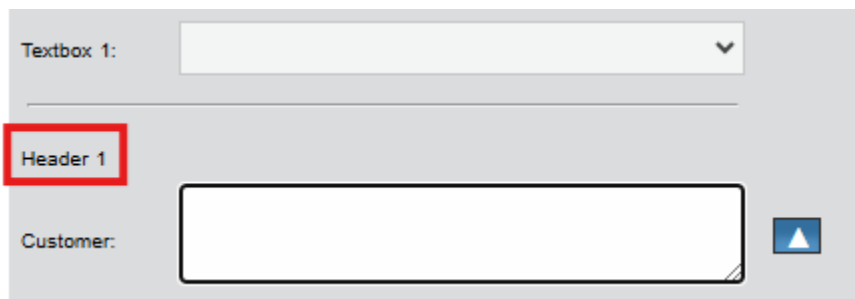
The screenshot shows a user interface with two main components. At the top, there is a dropdown menu labeled 'Textbox 1:' with a downward arrow icon. Below this, there is a section labeled 'Header 1' containing a text input field labeled 'Customer:' with an upward arrow icon to its right. A red rectangular box highlights the 'Textbox 1:' dropdown menu.

- Divider – This is a line used to represent a divide between fields. It is completely stylistic and has no overarching functionality.



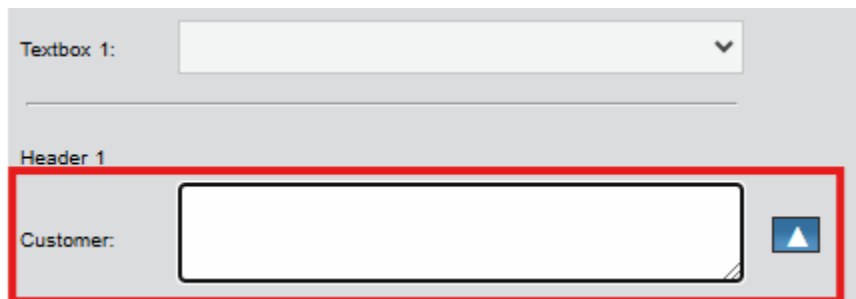
The screenshot shows a form with a 'Textbox 1' field at the top. Below it is a horizontal line, which is highlighted with a red rectangular box. Below the line is a 'Header 1' label. At the bottom is a 'Customer:' label next to a large text input field with a blue upward arrow button to its right.

- Header – This is a display only field that can be used to summarize or describe the current section of fields.



The screenshot shows a form with a 'Textbox 1' field at the top. Below it is a horizontal line. Below the line is a 'Header 1' label, which is highlighted with a red rectangular box. At the bottom is a 'Customer:' label next to a large text input field with a blue upward arrow button to its right.

- Text Area - Use this over the default 'textbox' field type when you are expecting longer / open-ended responses. This can be very useful on 'Note' fields where you want a user to enter more information than a textbox can display.



The screenshot shows a form with a 'Textbox 1' field at the top. Below it is a horizontal line. Below the line is a 'Header 1' label. At the bottom is a 'Customer:' label next to a large text input field, which is highlighted with a red rectangular box. A blue upward arrow button is to the right of the text area.

Thank you for reading along and I hope this brief article can help you add more value to your ScanWorkX implementation! If you would like to know more about customizing applications in ScanWorkX, have any questions/issues, or if you have an idea you think would be a beneficial addition to ScanWorkX please do not hesitate to reach out to us at harrison.brazelton@nextw.com.