Project Release Notes

Release 3.3 (March 2021)

What's new

Support for mandatory and read-only fields in lists

Editing in lists just got smarter! Set mandatory fields to flag errors when users submit records with blank fields and enable read-only fields to prevent users from changing the field value.

Read more...

Enhancements to child and grandchild sections

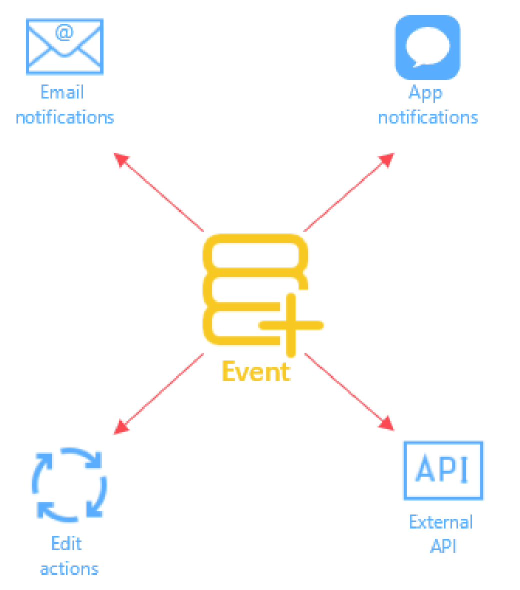
This release includes several clever tweaks to provide a smooth and easy user experience in viewing, editing, and creating records in child/grandchild sections. The list includes collapsable side panels, display-toggle, and quicker navigation for record editing.

Read more...

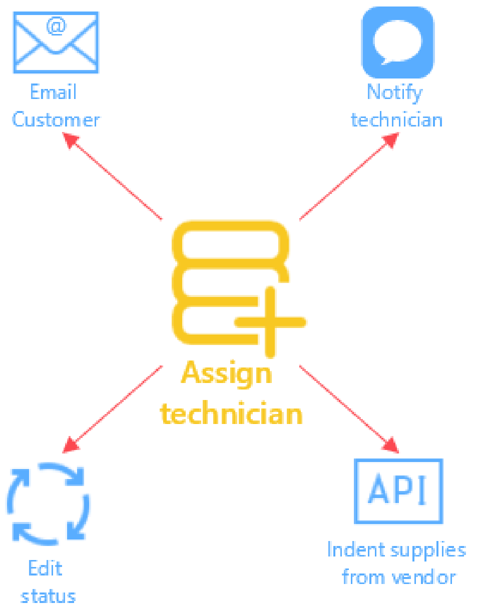
Save actions is now called Event-driven actions

We renamed the Save Actions tab to Event-driven Actions in the Studio’s Appify setup page. That is what the feature is all about - triggering actions based on the occurrence of specific events. You can trigger the following types of actions for an event:

- Send email notifications
- Send app notification
- Invoke silent edit actions
- make external API call



Configurable actions for an event



An example of an event: assign technician to work order

Event-driven actions - Edit actions

Enable your app users to breeze through the task of filling out forms. Using the new Edit actions feature in Event-driven actions (EDA), add an event rule to update records automatically.

For instance, you can configure a silent edit action to automatically update the current date in the ‘Closed on’ field in the event of users marking the Work order as completed. Similarly, add a rule to update the status field to ‘New’ for newly created Accounts.

Read more...

Event-driven actions - External API call

You can now automatically create and update records on external data stores, such as your partners' and vendors' information systems, as part of your workflow automation. Simply configure External API calls in your Event-driven actions (EDA).

For example, in a procurement app, for instance, when the app user approves a purchase order, you can now configure an external API call to directly place an order in the vendor's system using the REST API shared by them.

Read more...

Record uniqueness check

Ensure that a record entry in Flex is unique based on a combination of field values instead of just one. Include up to three fields to validate if their values when combined are unique across the object. Prevent record creation if the check fails.

For example, in the master list for Products, ensure that no two entries have the same product type, name, and model.

Read more...

Auto number subsequence

Enable auto-number subsequence to bind the generated auto-number value with a reference field of the Flex object.

For example, in Test orders, where you accept samples from several labs, to uniquely identify a sample from a lab, you can embed the Lab ID in the auto number sequence for the test order.

Read more...

Copy fields support in child sections

The copy fields feature is now supported in child sections too! If you include a reference field in a child section, you can set up the copy fields to automatically populate the child section's field with values from the reference record.

For example, in the Work orders section for an Account, you can automatically populate the phone number, email ID, and location details when the users choose a name from the contacts lookup.

Read more...

Support for USERINFO literals in Web client

We've added support in Web client for evaluating USERINFO group of literals and populating the field value based on the signed-in user's information. You can use USERINFO literals to set default values in all fields of type TEXT and TEXTAREA in create action screens.

For example, you can set the default value for the First name field in a create action screen using ${USERINFO.firstname}. This field will automatically be populated with the Web client user's first name when they open the form.

Read more...

Hide sections

You can now add hidden sections in the layout designer to store fields that are used in field calculations and validation rules computation but are not required to be displayed.

Read more...

Lists - Mandatory and read-only fields

Support for mandatory and read-only fields in lists

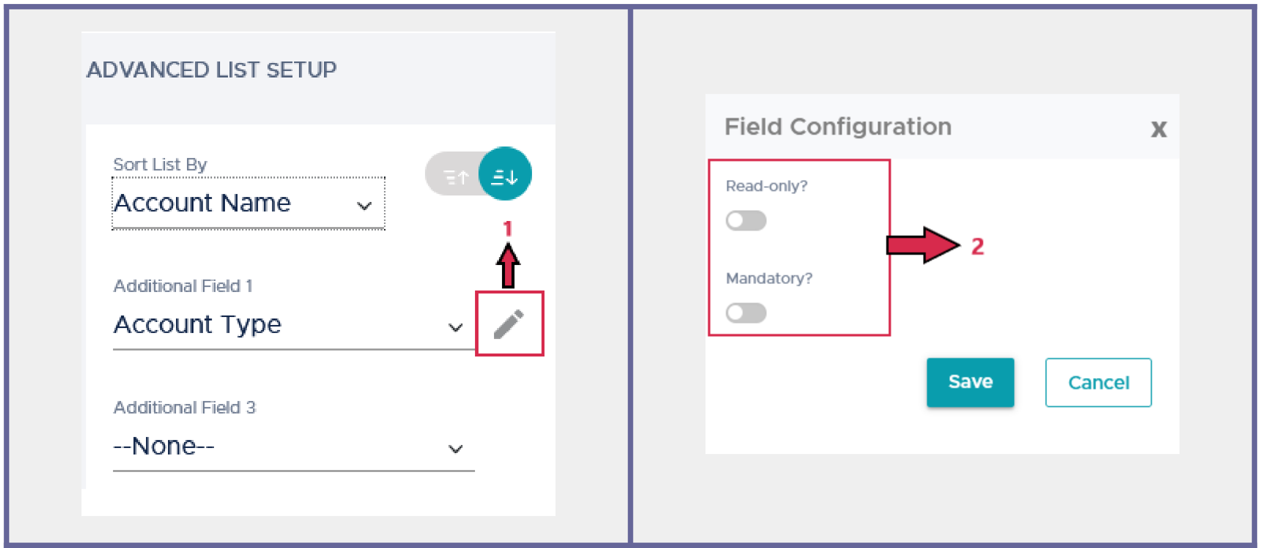
Editing records in lists just got smarter! Set mandatory fields to prompt app users when they submit records with blank fields and enable read-only fields to prevent users from changing the field value.

Enabling mandatory and read-only fields

You can now configure properties for list fields. Two properties, mandatory and read-only, are available for configuration. Enable the mandatory property to prompt app users to enter the field value when they click Save on the list page if the field is blank. Set the read-only property for a field to lock it on the app's List page and prevent users from changing the value.

By default, the mandatory and read-only properties inherit their values from the field configuration in the object. You can't disable these properties when it's enabled in the object.

To enable the mandatory' or read-only properties for a field, click on the pencil icon on the field's right and turn on the property from the 'Field configuration' pop-up that appears. See Figure 1 below.



Sort List By: Account Name

Additional Field 1: Account Type

Additional Field 3: --None--

Figure 1. Enabling mandatory and read-only fields

Support info

Supported Clients	Web client
Supported data sources	Flex and Oracle

Notes for app users

- The message “This field is mandatory. Enter a valid value” is displayed to app users if they click Save without entering any value in mandatory fields
- App users will not be able to edit the value in read-only fields

Enhancements to child and grandchild sections

This release includes several clever tweaks to provide a smooth and easy user experience in viewing, editing, and creating records in child/grandchild sections. The list includes collapsable side panels, display-toggle, and quicker navigation for record editing.

Users will see these improvements create, edit, and fork actions.

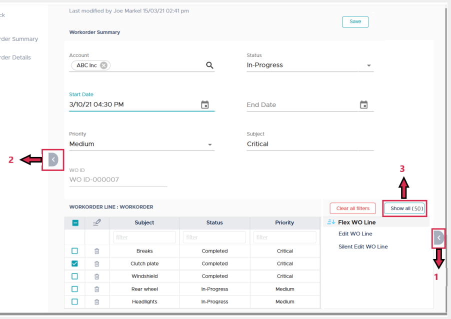
More room for displaying child records

Collapsable side panels

Users can click the collapse / expand icons on the child section's right (item 1 in Figure 1) and the section navigation panel's left (item 2 in Figure 1) to increase the display area for the columns.

Record display toggle

Using the Show all / less button (item 3 in Figure 1), users can display all the child section's records. Choosing 'Show all' will hide the other sections and show the chosen child section only on the screen. Clicking 'Show less' will take the users back to the normal mode.



Last modified by Joe Markel 15/03/21 02:41 pm

Workorder Summary

Account: ABC Inc

Status: In-Progress

Start Date: 3/10/21 04:30 PM

End Date

Priority: Medium

Subject: Critical

WO ID: WO ID-000007

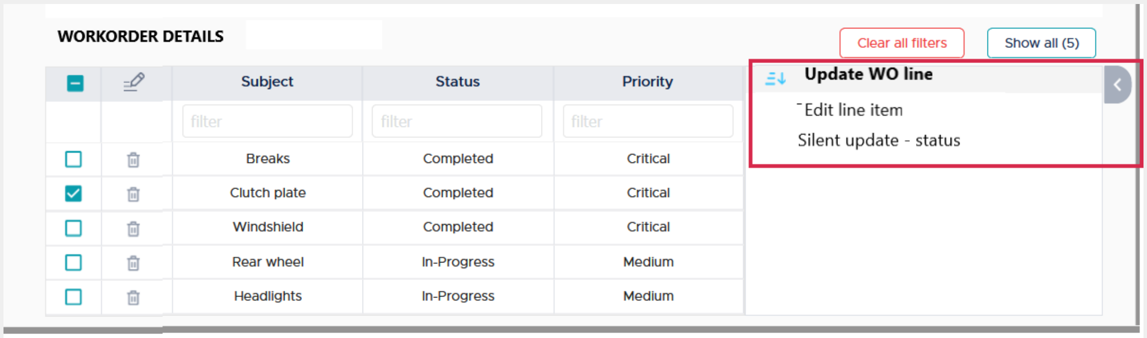
WORKORDER LINE : WORKORDER

		Subject	Status	Priority
□		Breaks	Completed	Critical
☑		Clutch plate	Completed	Critical
□		Windshield	Completed	Critical
□		Rear wheel	In-Progress	Medium
□		Headlights	In-Progress	Medium

Figure 1. The collapse panel icons and the Show all button

Improved record editing experience

- The record-level edit icon is gone; users don’t need it anymore. They simply have to double-click the field to edit a single record or use the checkbox to select records for editing.
- Users can now also access and execute silent edit actions from the actions panel for the selected records.
- We have done away with the record-level checkbox to save records. All inline edits child sections are saved when the users click the Save button for the section.



WORKORDER DETAILS

☐		Subject	Status	Priority
		filter	filter	filter
☐	🗑	Breaks	Completed	Critical
☑	🗑	Clutch plate	Completed	Critical
☐	🗑	Windshield	Completed	Critical
☐	🗑	Rear wheel	In-Progress	Medium
☐	🗑	Headlights	In-Progress	Medium

Update WO line

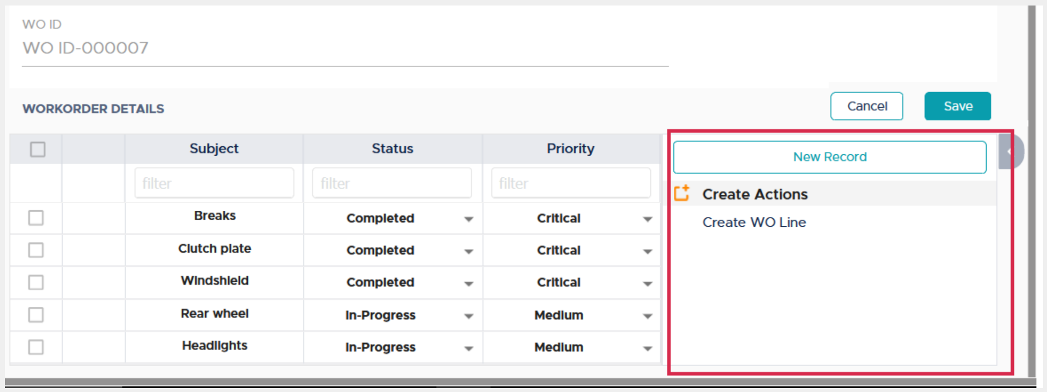
Edit line item

Silent update - status

Figure 2. Edit and Silent edit actions for child sections

Reorganized record creation tools

- We have moved The ‘New record’ button to the collapsable actions panel. Users can access the action panel by clicking the expand icon. See Figure 3.
- Users can either use the New Record button to add a new record directly to the child section or click on the required create action from the actions panel to open the record creation form.



WO ID

WO ID-000007

WORKORDER DETAILS

□	Subject	Status	Priority
	filter	filter	filter
□	Breaks	Completed	Critical
□	Clutch plate	Completed	Critical
□	Windshield	Completed	Critical
□	Rear wheel	In-Progress	Medium
□	Headlights	In-Progress	Medium

Figure 3. New Record button in the actions panel

Support info

Supported Clients	Web client
Supported data sources	Flex and Oracle

Event-driven actions - Edit actions

Enable your app users to breeze through the task of filling out forms. Using the new Edit actions feature in Event-driven actions (EDA), add an event rule to update records automatically.

For instance, you can configure a silent edit action to automatically update the current date in the 'Closed on' field in the event of users marking the Work order as completed. Similarly, add a rule to update the status field to 'New' for newly created Accounts.

Configuring edit actions for an event

Look for the new 'Edit actions' tab (item 1, figure 1) under the 'Actions to be performed' section in EDA's rule configuration page.

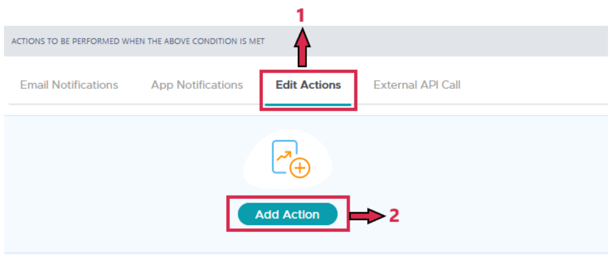
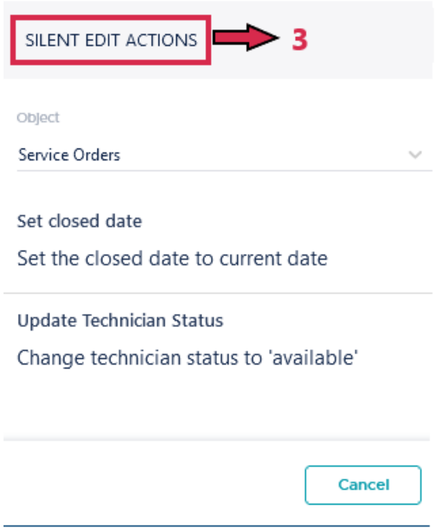


Figure 1. Edit actions tab and the Add actions button



Object

Service Orders

Set closed date

Set the closed date to current date

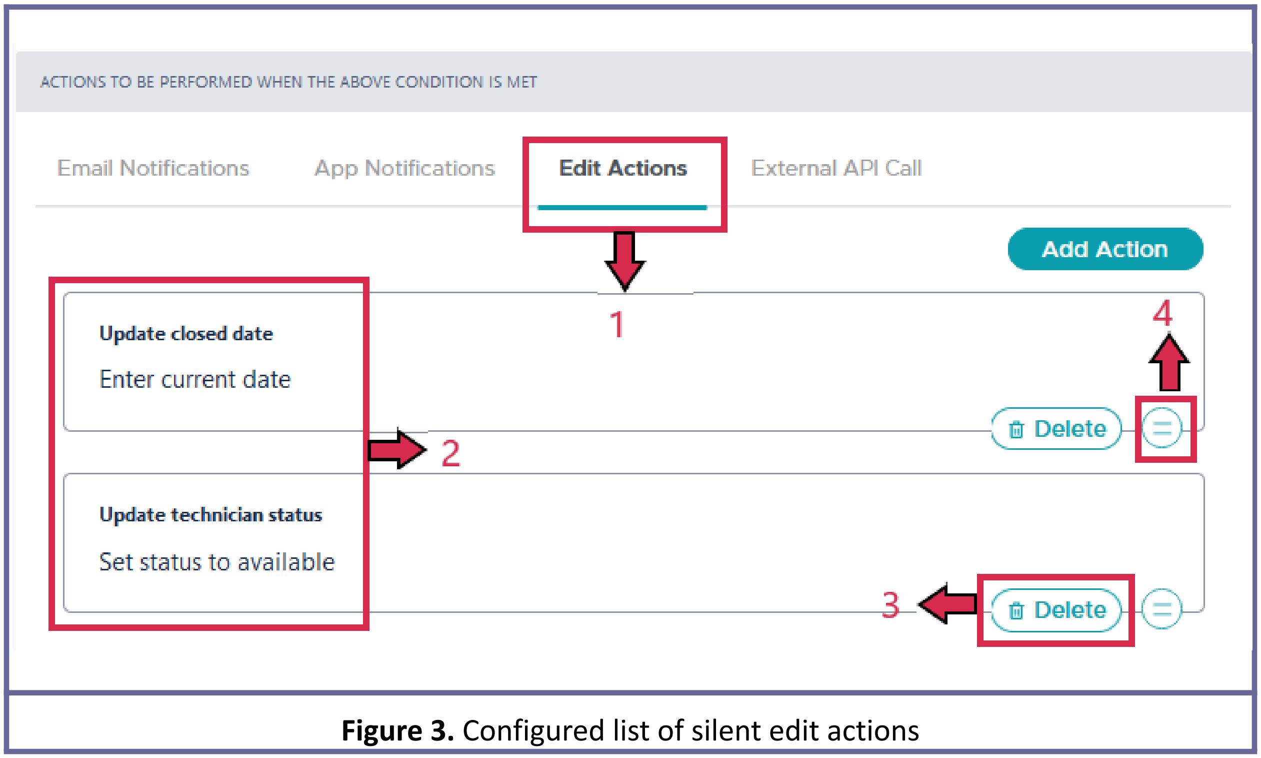
Update Technician Status

Change technician status to 'available'

Figure 2. The silent edit actions pop-over

To configure edit actions for the event, click 'Add actions' (item 2, figure 1) under the 'Edit actions' tab (item 1, figure 1).

The Silent edit actions pop-over (item 3, figure 2) that appears on the right lists the silent edit actions available for the chosen object. The object configured for the rule is chosen by default. Click the required action to add it to the list of actions to be triggered for the event.



ACTIONS TO BE PERFORMED WHEN THE ABOVE CONDITION IS MET

Update closed date

Enter current date

Update technician status

Set status to available

Figure 3. Configured list of silent edit actions

Adding edit actions from a reference objects

You can also trigger an update for a reference object when the event occurs. For example, you can choose to run any silent edit actions belonging to either the Work order object - configured for the event - or its reference objects such as Account, Location, and Installed Products.

To add a silent edit action for a reference object, click the ‘Objects’ drop-down and choose the required object to list its silent edit actions. Add the needed action by clicking it.

Defining the trigger sequence for the actions

When the event occurs, the actions are triggered in the same order they are listed in. To change the sequence, click the reorder icon (item 4, figure 3) and drag and drop the action in the required order.

If a silent edit action fails during execution, it will be skipped, and the next action on the list will be triggered.

Support info

Supported Clients	Web client
Supported data sources	Flex and Oracle

Notes for app users

Based on the silent edit actions you add in the EDA configuration, the form fields are automatically updated for the app users when the event and the edit action trigger criteria are met.

Event-driven actions - External API call

You can now automatically create and update records on external data stores, such as your partners' and vendors' information systems, as part of your workflow automation. Simply configure External API calls in your Event-driven actions (EDA).

For example, in a procurement app, for instance, when the app user approves a purchase order, you can now configure an external API call to directly place an order in the vendor's system using the REST API shared by them.

Similarly, in a test request app for medical diagnostics, when the app users confirm a time slot for sample collection, you can directly book an appointment on the external lab's appointment booking system using their exposed REST API.

Configuring an external API call for an event

You can configure an external API for an event rule by adding the REST endpoint in the newly introduced 'External API call' tab. You can configure the following details for the external API call:

- API endpoint where the service is available
- HTTP Method to use
- Additional HTTP headers that you need to send
- Custom data that you need to include in the request body
- Authentication details
- Field values that you need to update in the external datastore

Please reach out to your Appify contact for help with configuring external API calls in event-driven actions.

Support info

Supported Clients	Web client
Supported data sources	Flex and Oracle

Notes for app users

When you configure an external API call for an event, the associated external data store is automatically updated when the event occurs. The app users no longer have to deal with updating dependent external records.

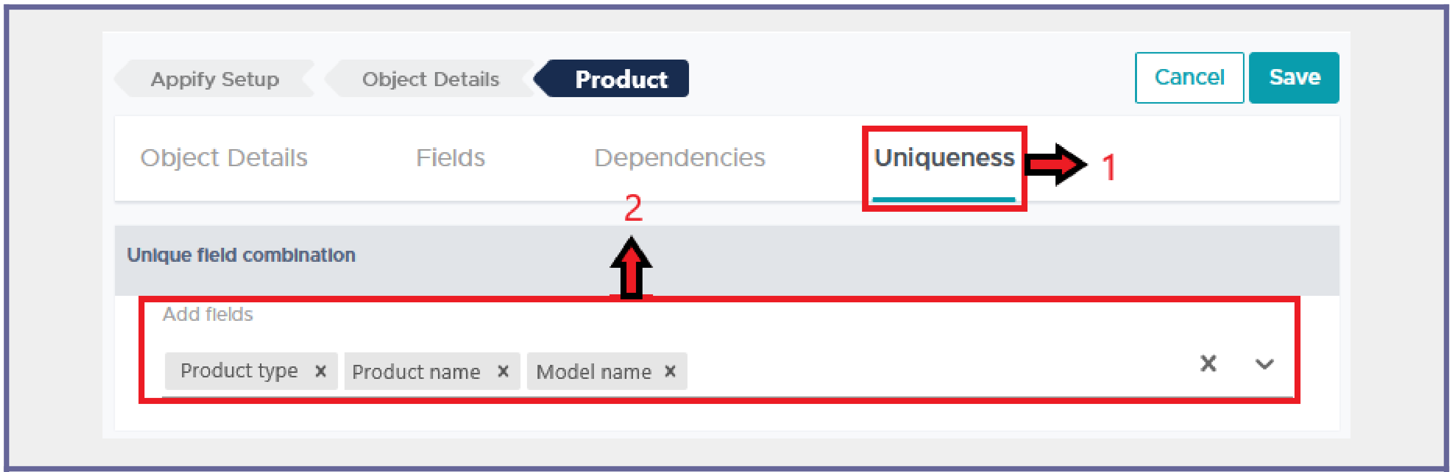
Record uniqueness check

Ensure that a record entry in Flex is unique based on a combination of field values instead of just one. Include up to three fields to validate if their values when combined are unique across the object. Prevent record creation if the check fails.

For example, in the master list for Products, ensure that no two entries have the same product type, name, and model.

Setting up record uniqueness check

Look for the new 'Uniqueness' tab (item 1, figure 1) on the object configurations page. To add a combination of unique fields for a Flex object, click the 'Uniqueness' tab (item 1, figure 1) and select the fields from the 'Unique field combination' drop-down (item 2, figure 1). You can select up to three fields for uniqueness validation.



Unique field combination

Add fields: Product type, Product name, Model name

Figure 1. Record uniqueness configuration page

Support info

Supported Clients	iPad, iPhone, Android, and Web client
Supported data sources	Flex

Notes for app users

When you configure record uniqueness check for Flex objects, an error message is displayed to the users if they try to save a record with duplicate values in unique field.

Auto number subsequence

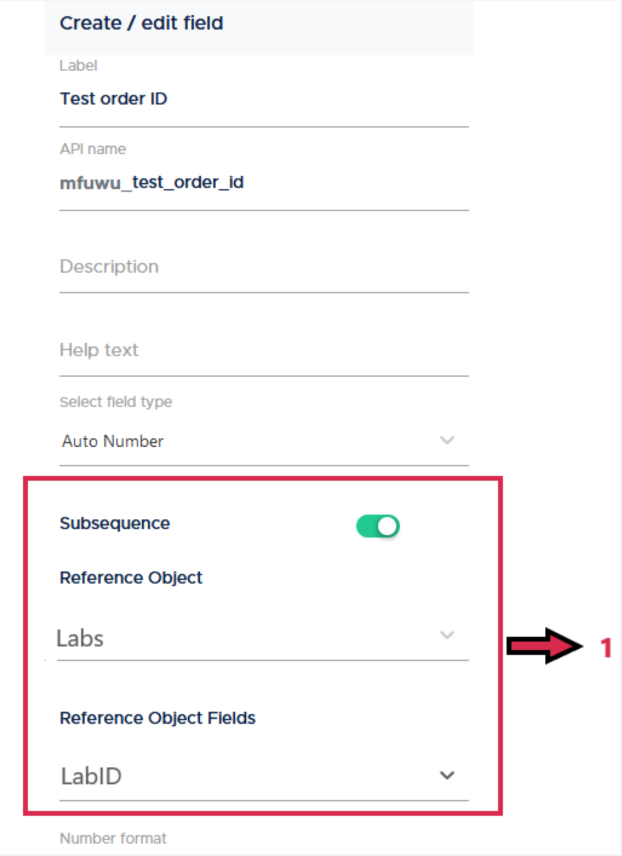
Enable auto-number subsequence to bind the generated auto-number value with a reference field of the Flex object.

For example, in Test orders, where you accept samples from several labs, to uniquely identify a sample from a lab, you can embed the Lab ID in the auto number sequence for the test order.

Enabling subsequence for auto numbers

You will see a new toggle named ‘Subsequence’ (item 1, figure 1) in the fields configuration panel for auto-number fields. Subsequence is disabled by default.

To enable subsequence for an auto number field, turn on the toggle, select the required reference object and field. Only reference objects corresponding to the reference fields used in the object are listed in the ‘Reference object’ drop-down.



Create / edit field

Label
Test order ID

API name
mfuwu_test_order_id

Description

Help text

Select field type
Auto Number

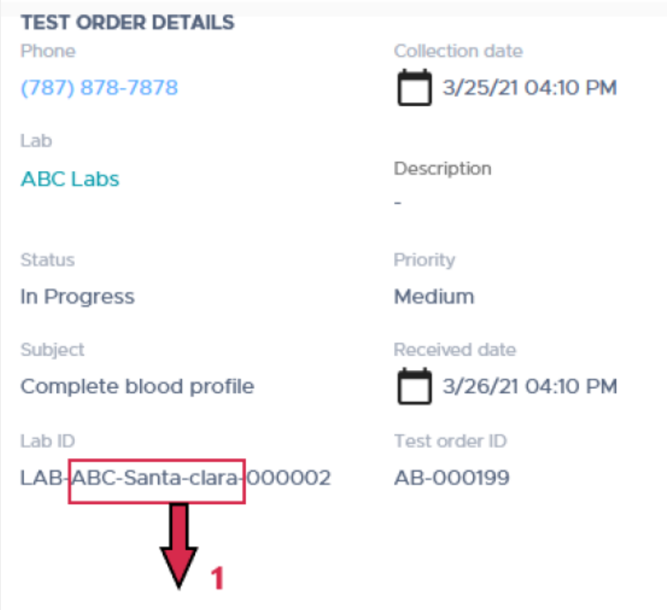
Subsequence

Reference Object
Labs

Reference Object Fields
LabID

Number format

Figure 1. Enabling subsequence for an auto number field



TEST ORDER DETAILS

Phone
(787) 878-7878

Collection date
3/25/21 04:10 PM

Lab
ABC Labs

Description
-

Status
In Progress

Priority
Medium

Subject
Complete blood profile

Received date
3/26/21 04:10 PM

Lab ID
LAB-ABC-Santa-clara-000002

Test order ID
AB-000199

Figure 2. Auto number field with subsequence - embedded lab ID

Note:

- Newly applied subsequence will not apply to existing record
- If the reference field is blank at the time of generating the auto number, it will not include any subsequence value
- Once generated, the value of the auto number will not change if the reference field value changes or the field itself is deleted at a later point

Support info

Supported Clients	iPad, iPhone, Android, and Web client
Supported data sources	Flex

Notes for app users

When you enable subsequence for an auto number field, the generated ID value will include the corresponding reference field’s value (item 1, figure 2) in the new record created by the app users.

Copy fields support in child sections

The copy fields feature is now supported in child sections too! If you include a reference field in a child section, you can set up the copy fields to automatically populate the child section’s field with values from the reference record.

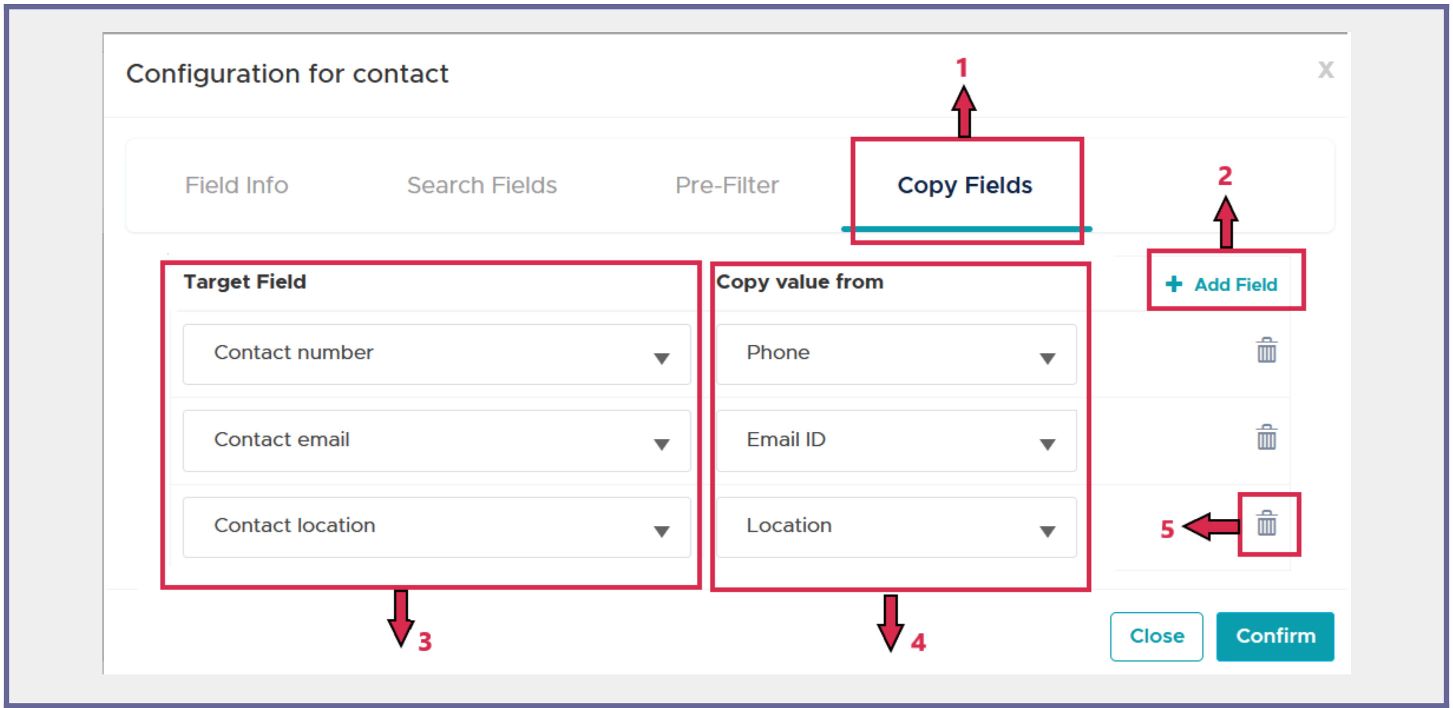
For example, in the Work orders section for an Account, you can automatically populate the phone number, email ID, and location details when the users choose a name from the contacts lookup.

Copy fields configuration was only supported in the standard sections, with this release it is supported in child sections too.

Setting up copy fields for child section

The ‘Copy fields’ tab (item 1, figure 1) appears in the field configuration window of the reference fields in the child section. To set up copy fields, use the copy fields tab, click ‘Add fields’ (item 2, figure 1), and from the ‘Target field’ drop-down (item 3, figure 1), choose a field. Note that only fields included in the child section are listed in the target fields drop-down.

Map the target field to the reference object’s field as needed, choose a field from the ‘Copy value from’ drop-down (item 4, figure 1). Note that only fields that match the chosen target field’s data type are available for mapping. You can delete the mapped pair of fields by using the bin icon (item 5, figure 1) on its right.



Target Field	Copy value from
Contact number	Phone
Contact email	Email ID
Contact location	Location

Figure 1. Setting up copy fields for child section

Support info

Supported Clients	Web client
Supported data sources	Flex

Notes for app users

The configured target fields in the child section are automatically populated using the values from the record corresponding to the lookup value chosen by the app users.

Support for USERINFO literals in Web client

We’ve added support in Web client for evaluating USERINFO group of literals and populating the field value based on the signed-in user’s information. You can use USERINFO literals to set default values in all fields of type TEXT and TEXTAREA in create action screens.

For example, you can set the default value for the First name field in a create action screen using ${USERINFO.firstname}. This field will automatically be populated with the Web client user’s first name when they open the form.

List of available ${USERINFO} literals

This is the complete list of ${USERINFO} literals you can use:

Literal	User information returned by the literal
${USERINFO.firstname}	First name
${USERINFO.lastname}	Last name
${USERINFO.phone}	Phone number
${USERINFO.email}	Email address
${USERINFO.fullname}	Full name
${USERINFO.companyname}	Company name
${USERINFO.city}	City
${USERINFO.street}	Street

${USERINFO.state}	State
${USERINFO.country}	Country
${USERINFO.postalcode}	Postal code
${USERINFO.usertype}	User type
${USERINFO.title}	Title

Support info

Supported Clients	Newly added for Web client in this release. Already supported on iOS and Android apps.
Supported data sources	Flex
Supported actions	Create
Supported field types	TEXT and TEXTAREA

Notes for app users

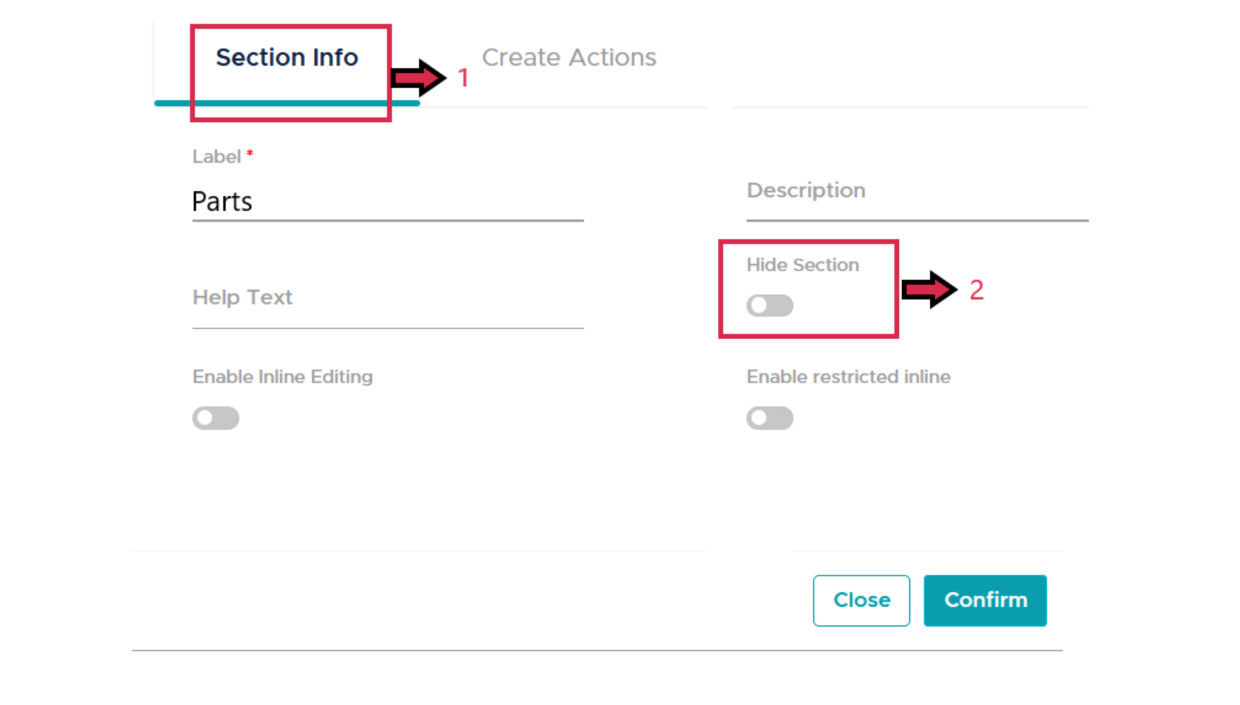
The configured target fields in the child section are automatically populated using the values from the record corresponding to the lookup value chosen by the app users.

Hide sections

You can now add hidden sections in the layout designer to store fields that are used in field calculations and validation rules computation but are not required to be displayed.

Hiding sections

A new toggle named ‘Hide section’ (item 2, figure 1) will now appear in the Section info tab (item 1, figure 1) of the section configuration window. This is available in all types of sections of create and edit actions. To hide a section, turn on the ‘Hide section’ option, it is disabled by default.



Label * Parts

Description

Help Text

Hide Section ○

Enable Inline Editing ○

Enable restricted inline ○

Figure 1. Hiding sections

Support info

Supported Clients	Web client
Supported data sources	Flex

Notes for app users

Hidden sections don’t appear on the screen when the app users open them.

Fixed issues

The following customer issues are fixed in this release:

Issue ID	Product area	Issue description
CD-792	Android	The ‘Add new’ icon to create child records still appeared in the Edit action’s child section when there were no create actions configured.
CD-1194	Studio	In the expression builder, all field types available in the main object were not available for selection in the subquery dropdown.
CD-1235	Web client	The ‘Hide when’ and ‘Enable when’ action criteria didn’t evaluate correctly in the app if the filter expression included any USERINFO literals. The actions appeared/were enabled even when the condition matched.
CD-1236	Web client	The number of records displayed per page in a list was not as per the configured value; ten records were displayed per page.
CD-1240	Web client	App users had to click the save button unnecessarily when they ran any bulk actions in a list even though the bulk actions automatically save the changes.
CD-1247	Web client	Reference fields marked as read-only in the list were available for editing in the app.

CD-1248	Web client	If any mandatory fields were blank in a list when users executed bulk action, the validation check did not report the error. However, it did not save the changes made by the bulk action.
CD-1249	Web client	The changes made to records using bulk action were not saved if there were any read-only fields in the list.
CD-1286	Studio	The fields added newly to the Oracle objects didn’t appear on the Object modeler page.
CD-1377	Web client	Saving changes in a newly added edit action caused an error. It was not possible to save the changes due to this.
ENG-3357	Android, iOS	In Appify mobile apps, during data sync, if the users opened any other app, the sync stopped. The app had to be restarted for data sync to continue.