

InfoCapture best practices: Administrator top tips

InfoCapture is a substantial and powerful tool that can manage your information processes effectively when used correctly.

We appreciate there is a lot to get to grips with in InfoCapture and its particular mechanisms will become familiar with the experience.

Below are some best practice tips for administrators.

- Designate responsible users for your forms
- Keep the number of project roles in a form to a minimum
- Revert form field changes easily by deleting the latest form version
- Be aware that form logic is not tied to the form version (only form fields are)
- Add a version comment at every check-in to track form field changes effectively over time
- Remain vigilant if performing data imports
- Maintain a separate test version of a form (or a general form used for testing)

- Designate responsible users for your forms

Select form managers who will be responsible for administering and managing certain forms, as well as their changes.



For established forms, any changes made need to be well informed to prevent issues from being created as a consequence.

These individuals (or teams) need to become experts on the forms they look after e.g. how they work, what their purpose is, who needs to access them, how critical the form is to the company and crucially how to make changes if required.

When there is an issue in InfoCapture your responsible users are best placed to investigate first and either resolve using their own knowledge or submit a ticket with all relevant information they have.

These representatives can take our course to get started on form building, watch our webinars and follow all our literature to gain further understanding.

We at Claromentis are on hand to assist in creating a form or offer advice on how to make changes, however, it's always best to come from an internal team who understands your company and its needs.

If these users can build the form or become an expert on those which are needed day-to-day they are absolute assets to your company as they can problem solve in the first instance.

- Keep the number of project roles in a form to a minimum

An ideal number of roles would be no more than 3. e.g.

- **Administrators** - All rights, able to manage every part of the form as needed

- **Manager/role that fits your forms higher purpose** - More rights to interact with tickets as required for the forms use case
- **Ticket submitters** - Least rights, likely to be end users only

Admin / Infocapture / Support Tickets / Project permissions

Project options

Project Summary

Edit project properties

Project permissions

Statuses

Conditions

Field condition sets

Triggers

Behaviour

Field visibility

SLA

Automatic changes

Project permissions

Create project roles and define their permissions within the project.

User groups, roles, and individuals can be put into project roles. Each role has its permissions within the project defined here. One role may be given rights to only submit tickets, and a higher role may be given rights to update and assign them, for example. Additionally users can be granted additional rights beyond what their project role would ordinarily allow, just for tickets they have personally submitted and those assigned to them.

Project roles

Rights

Options

All

+ Add new project role

	Role name	Users		
<input type="checkbox"/>	Admin	Role: Administrators		
<input type="checkbox"/>	Support	Group: Support		
<input type="checkbox"/>	Users	All registered		

Delete selected

As a general rule, if you are creating more than 5 roles, it’s likely that you are overthinking and adding unnecessary complexity.

The more complex the form, the more responsible users will need to be aware of its intricacies and how they work.

It’s understandable you may need differentiation between roles, but try to assimilate some of these to keep the number of roles below 5 and make managing the form easier.

Less is more!

- Revert form field changes easily by deleting the latest form version

If a change is made to a form field that was a mistake or adversely affected your form once made live, the quickest way to rectify this is to delete the form version containing the change.

Rather than making further changes to a form to rectify the issue and checking this in, it is better to remove the form version and then build onwards freshly from this.

Admin / Infocapture / Meetings / Delete form version

Meetings

Warning: The following number of tickets were created with this version of the form

Date of first filling

01-09-2022 15:45

Date of last modification

28-09-2022 09:53

What would you like to do with the tickets that use this form?

☐ Reassign to the next version of form

☒ Reassign to the previous version of form

☐ Delete

Cancel

Delete

Hopefully, this would have been caught quickly so will be the most recent form version checked in and there will be no new tickets submitted in it. If so, it can be deleted straight away, or it has tickets submitted in it and these can be attributed to the previous form version when the most recent is deleted.

(Or those tickets could also be deleted and then resubmitted once the version is removed, if appropriate)

- Be aware that form logic is not tied to the form version (only form fields are)

Any changes made to the fields that make up a form are required to be saved in a new version of that form and checked in.

Admin / Infocapture / Bug Tracker / Check in form

Check in form

Version comment

How to use this version

☒ Only for new ticket

☐ Replace the previous version with this one

☐ Replace all previous versions with this one

Save

A new version can be applied to:

- Only new tickets
- Tickets submitted in the previous version only (and new tickets going forward)
- All tickets ever submitted in the project (and new tickets going forward)

The choice affects past, present and future tickets as the form version they are saved under is being altered.

Outside of forms fields, every other change to the form (i.e. status, field conditions, field visibilities, notifications etc.) will always apply when updated as these are not tied to the form version.

This is something for responsible users to be aware of to avoid any confusion over what is being changed and when as well as what it will impact.

e.g. Making a change to the statuses within a form will be immediately applied and appear on the front-end. The only way to revert changes or further modify them is manually from the admin side of the form vs making a change to form fields, which is tied to the version checked in and the recommended way to revert this is to delete the form version where they exist.

- Add a version comment at every check-in to track form field changes effectively over time

When checking in updates to your form fields...

The screenshot shows the 'Employee Of The Month' form editor. At the bottom left, the 'Check in form' button is highlighted with a red box. The form has a 'Nomination' section with a 'Reason' text area. A red box also highlights the 'Discard all changes' button at the bottom right.

...ensure a comment is left outlining what was changed at this stage:

The screenshot shows the 'Check in form' dialog box. The 'Version comment' field contains the text: 'Updated the People groups for the 'nomination' field - from 'All Registered' to 'Company+'. The 'How to use this version' section has radio buttons for 'Only for new ticket', 'Replace the previous version with this one', and 'Replace all previous versions with this one'.

This is paramount if any issues arise with the form in the future, the potential for this to be linked to a certain form version can be ascertained easily as each version will have a comment on what was updated.

Meetings					
Form versions					
View form	Version comment	Number of tickets	Modified	Modified by	Delete
View	Decisions field added	8	10-03-2015 15:25		Delete
View	Action Recipients field added	0	10-03-2015 14:28		Delete
View	Form created	0	10-03-2015 13:52		Delete

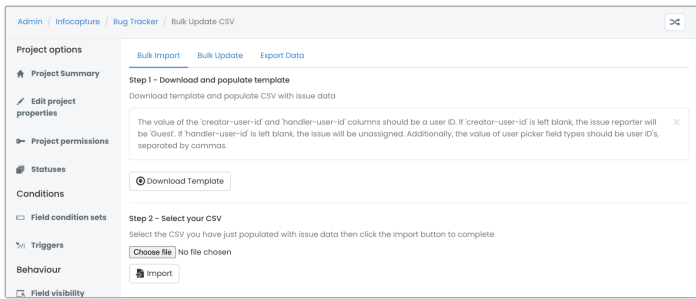
Without a version comment, it's difficult to be sure what was added or removed when (without trawling through the archived audit logs), especially if a form is complex or previous responsible users in charge are no longer at the company.

- Remain vigilant if performing data imports

When importing ticket information to a form it is important to ensure the data does not get disorganised, especially ticket IDs which should remain static.

Only run an import once and if an error is encountered investigate this first before attempting the import again as this can cause duplication which can be difficult to unpick in the database.

If you encounter an error on import you cannot resolve yourself, let us know in a support ticket and we can assist before another import is run to minimise the impact on your data.



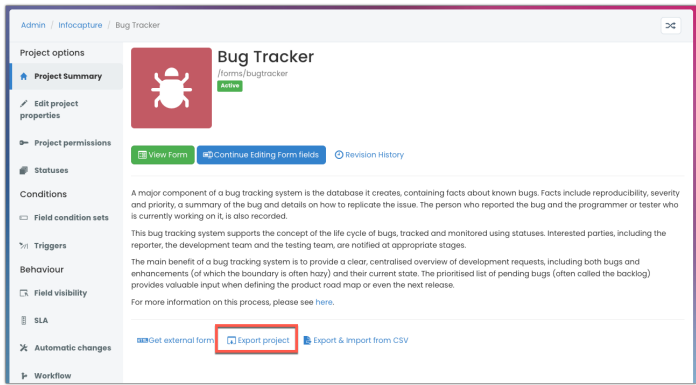
- Maintain a separate test version of a form (or a general form used for testing)

A common way to ensure any changes you wish to make to a live form have been tested ahead of implementation is to have a test copy of it on your site.

This can include just your responsible users in its permissions so only they can see and interact with it.

The risk of carrying out changes to the test form is far lower and can be more easily rectified as no end users are actively relying on it working for their jobs.

Whole forms can be exported to ensure you always have an up-to-date copy to import when needed for testing.



Recommended next article:

Form functionality changes

Related Article

[Form functionality changes](#)

[InfoCapture investigative tools](#)