

Sapphire Guide

Document version: 3

Prepared for Sapphire version 1.2.0

07 May 2020



Copyright 2020 Axiell

All rights reserved. No part of this work may be copied, reproduced and/or transmitted in any form by any means, electronic or mechanical, including photocopying and recording, or by any information storage or retrieval system, except as may be permitted in writing by Axiell.

CONTENTS

Overview	1
Two broad functions	1
Types of users	2
Version and upgrade information	3
Important: upgrading to Sapphire 1.2.0	3
Working with Sapphire	5
Logging in / out	5
Data Entry user	7
Working with a Form	8
Working with images and other multimedia	12
Uploading multimedia files	14
Polygons	15
Attaching and viewing images	17
How to search for and edit a record	20
How to use a Sapphire Editor to clean up data and process attachments	22
Example use case	22
How to use an Editor	23
Example: creating attachments	28
Programme Lead	32
A Form to capture or edit one record at a time	33
Perform some special processing	36
Logging in a Programme Lead	37
How to create a project and make it available to users	39
Registry entries	40
Name the project	40
Determine who has access to what for each project	41
Define a user's role	42
Define which projects a user has access to	43
How to create and edit a data entry Form	44
Kind: multiple	57
Type: date	59
Type: selection	61
Type: text	63
Lookup	65
Attachment field	67
Multimedia fields	70
The Deferrable system	82
How to customise the layout of a Form	83
Customising the layout of a data entry Form	85
How to specify permissions for project elements	89
How to add a Search box in a Form	91
Using a filter without including a Search box	96
How to create and edit an Editor	98
Examples of Editors	103
How to display Statistics	108
Manager	111
Minimum System Requirements	115
Release Notes	117
INDEX	119

Overview

Sapphire is a browser-based add-on to the EMu Collections Management System. It is designed to provide access to your EMu database for the creation of new records and editing of existing records through custom-built forms. A Form is designed and constructed by authorised users around a workflow, capturing data, editing records, creating attachments, etc., with each Form providing access to a subset of fields.

Features of Sapphire include:

- » Interfaces for desktops and tablets, providing a flexible working environment.
- » Automatic and continual live update of EMu, coupled with live validation with EMu data during capture.

Sapphire offers a seamless and transparent integration with EMu, and records are added to your EMu database immediately. Manipulation and enhancement of data is performed directly within EMu, ensuring that data capture / cleansing projects feed directly into the core data repository continually, and that all work performed is immediately available from that repository.

- » For authorised users, graphs are available for a range of real-time statistical overviews (page 111). Graphs can be designed to display any sort of statistic; a large scale digitisation project might include graphs that track the progress of the project, for instance.

Two broad functions

Sapphire is designed to perform two broad functions:

- » Data Capture: Forms

Forms are designed for a specific data capture purpose. A Form includes a subset of fields organised to capture a particular workflow (recording a new object, for instance). In a digitisation project in a Natural History museum, for example, Forms would be tailored for the Herbarium digitisation programme, the Palaeontology digitisation programme, and so on, each one designed to capture data relevant to each discipline.

- » Data Cleansing: Editors

Authorised staff are able to clean up / update data.

Data Cleansing can be as basic as standardising a value, i.e. replacing variations of a value with an approved value. A Sapphire Editor can also be used to create attachments. For example, during data entry a collector's name may have been captured from specimen labels and stored as text in a Catalogue record for the specimen; an Editor can be used to create an attachment from one or more Catalogue records to a Parties record for the collector.

See *How to use an Editor to clean up data and process attachments* (page 22) for details.

Types of users

Sapphire is browser-based and accessed via a secure login. Three types of user are envisaged:

1. Data Entry
2. Programme Lead
3. Manager

A user's type, which projects the user is assigned to (a user can be assigned to one or more projects) and which tools the user can access for each project are defined in the EMu Registry (page 39).

A project typically comprises:

» One or more Forms

Forms are designed by Programme Leads for a specific data capture / editing purpose. A Form includes a subset of fields organised to capture a particular workflow (recording a new object, for instance).

» Editors

Editors can be provided for data cleansing and processing data. A project may have one or more Editors.

» Statistics

Graphs presenting various statistics, such as the progress of the current project.

See *Manager* (page 111) for details.

Typically:

- » Data Entry users are presented with Forms when they log in to Sapphire. If authorised, they may have access to one or more Editors.
- » Programme Leads are presented with the Form, Editor and Statistics configuration tools.
It is the Programme Lead's responsibility to create projects, assign access permissions and make projects available to users (page 39).
- » Managers are presented with project statistics.

Version and upgrade information

This document was written for Sapphire version 1.2.0. It was published in May 2020 as a PDF and is also available on the EMu Online Help (help.emu.axiell.com).

Updates will be made to the EMu Online Help.

Important: upgrading to Sapphire 1.2.0

Upgrading Sapphire from any prior version of Sapphire to 1.2.0 will require some maintenance of existing Forms:

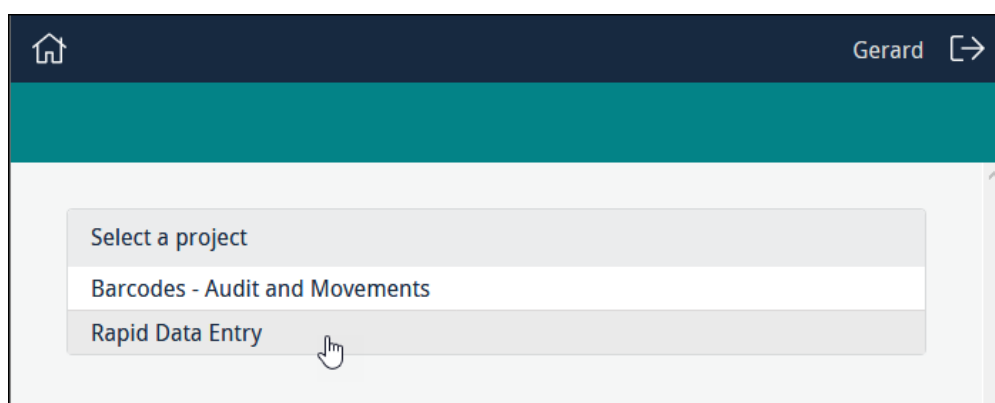
- » Re-save your Forms in the Form Designer.

In some earlier versions of Sapphire, sections in a Form were saved without a unique ID. As a consequence, sections may be missing from a Form or the Form will simply be blank following an upgrade to 1.2.0.

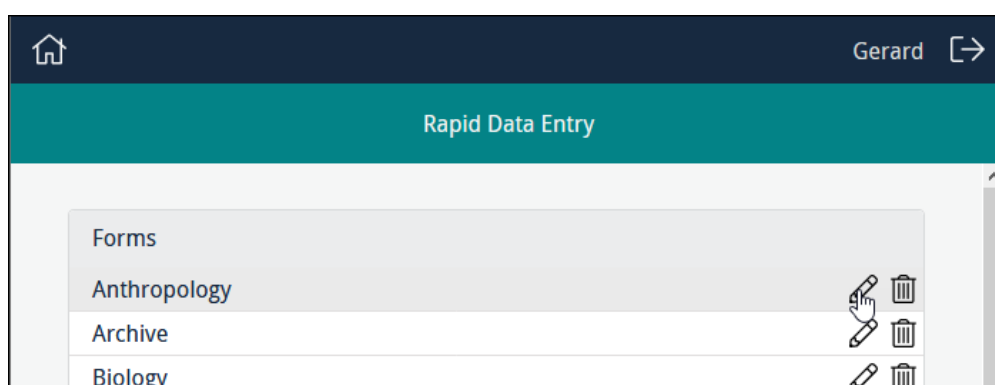
IDs are now automatically generated when a Form is saved in the Form Designer and we recommend that all Forms are opened in the Form Designer and re-saved.

This is a relatively simple process:

1. Select a project from the Dashboard:



2. Click  to edit a Form:



3. In the Form Designer, click **Save**.

» Rebuild any custom layouts.

Any Form with a custom layout (see page 85 for details of custom layouts) will need to be rebuilt as the old format is not compatible with Sapphire 1.2.0.

Although a Form with a custom layout designed in an earlier version of Sapphire will be usable, the elements on the Form will be laid out sequentially rather than as designed, and it will be necessary to rebuild the custom layout.

Working with Sapphire

Logging in / out

The Programme Lead creates projects, assigns access permissions and makes projects available to users.

Navigating to the URL provided by a Programme Lead will display a Login box:



Note: The Login form recalls the last username entered.

1. Enter your username and password.
2. If you are a member of more than one EMu user group, select the group you wish to log in as.

Projects, and the elements in a project such as Forms and Editors, can be assigned to EMu users and / or groups.

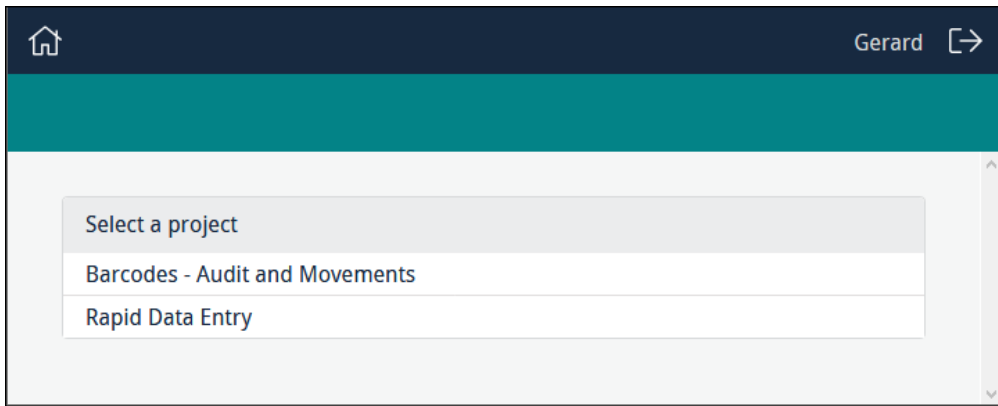


Note: If you do not select a group from the *Group* drop list, you will be logged in to the first group listed in the drop list by default and the permissions assigned to this group will apply.

3. Click **Log in**.

If you are a member of only one project, you will be presented with the project elements you have been authorised to work with (Forms, Editors, Statistics).

If you are a member of more than one project, a list of projects will display:



4. Click the name of the project you wish to work with.

A project comprises:

- » One or more Forms

- » Editors

Used to clean up and process data.

- » Statistics


Graphs presenting various statistics, such as the progress of a project.

Depending on your role, you will have access to one or more of these components, which will display in the following order:

- » Forms

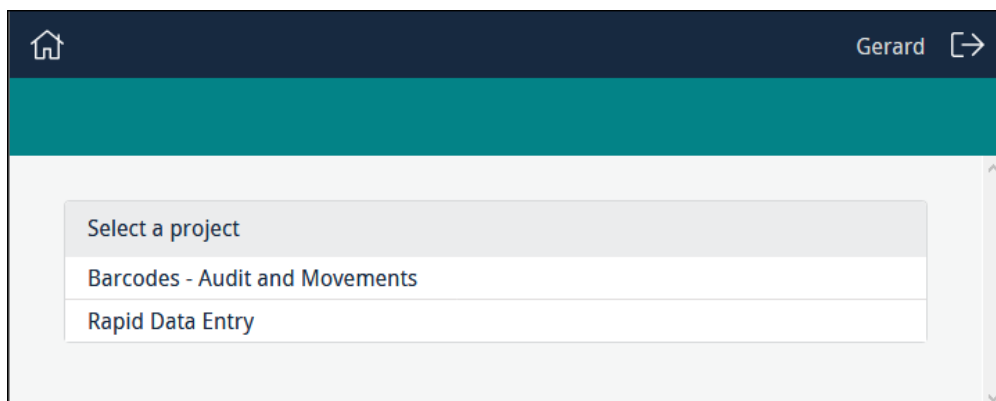
- » Editors

- » Statistics

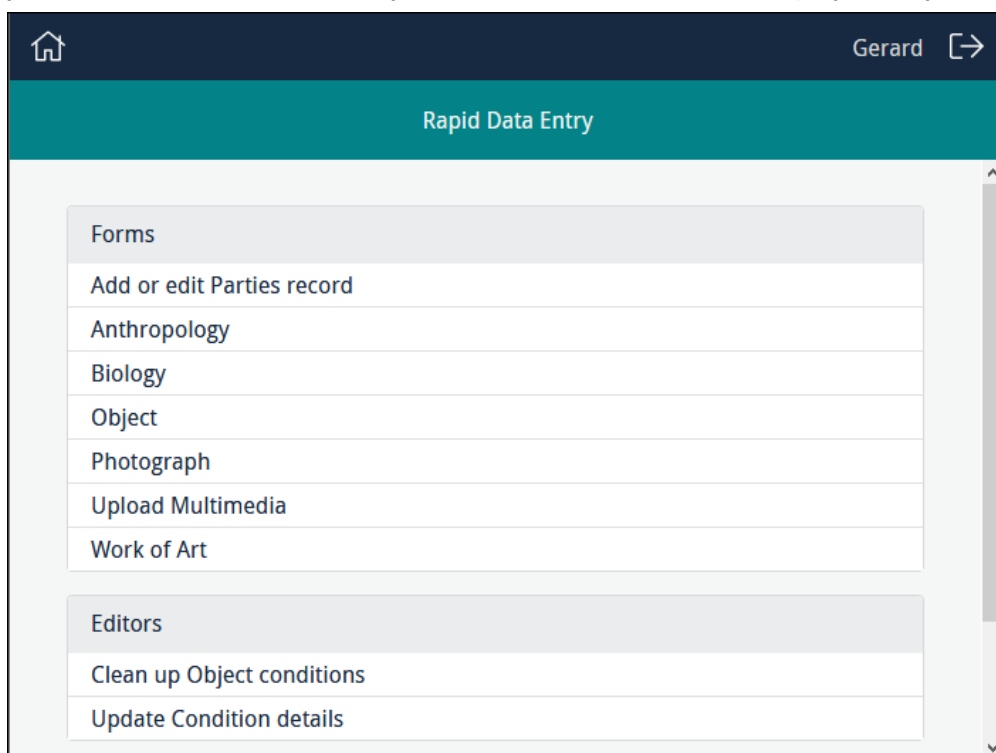
5. Click  to log out of Sapphire.

Data Entry user

If you have been assigned to more than one project, a list of your projects will display when you log in. Select the project you wish to work with:



If you have been assigned to one project, or you selected a project from the list of projects assigned to you, a list of Forms and Editors you are authorised to use will display when you log in:



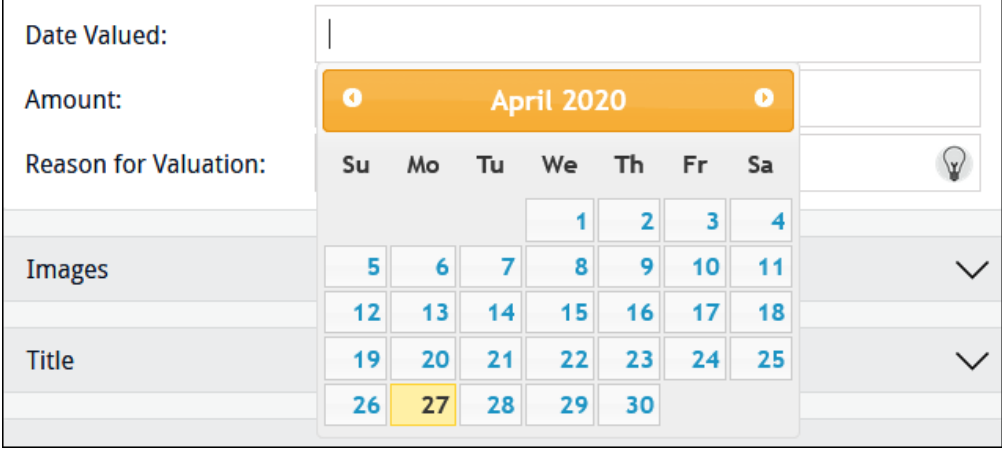
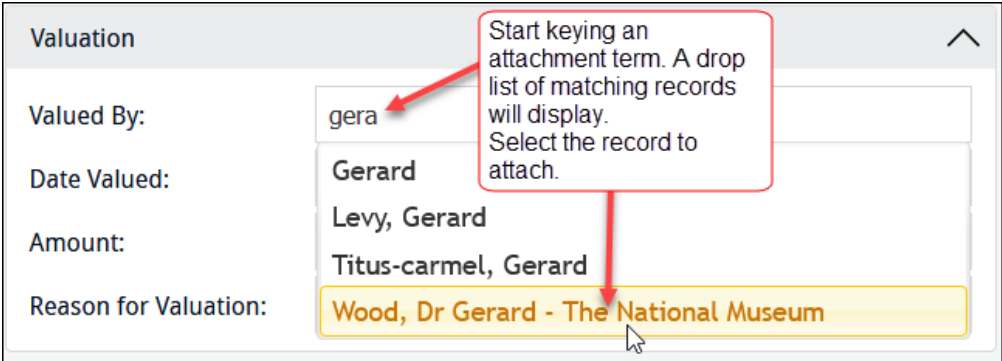
Click the label of the Form you wish to work with, e.g. **Object**, to display the Form ready for use.


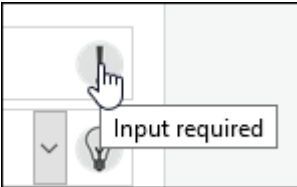



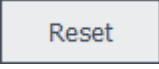



Working with a Form


Working with Forms is fairly straightforward and in most cases you simply key data into a field. Some fields are interactive and respond as data is entered into them, date and attachment fields for instance:

The screenshot shows a web form interface for an 'Object'. At the top, there is a dark blue header with a home icon, a folder icon, and the name 'Gerard' with a right-pointing arrow. Below the header is a teal bar with the word 'Object' centered. The main content area is a scrollable list of sections, each with a title and a chevron icon on the right. The sections are: an empty section with an upward chevron; 'Accession' with an upward chevron, containing a field 'Accession Lot:' with the value 'Old Parliament House (Donation)'; 'Location' with a downward chevron; 'Condition' with a downward chevron; and 'Valuation' with an upward chevron. The 'Valuation' section contains four fields: 'Valued By:' with an empty field and a warning icon (!); 'Date Valued:' with the value '(e.g. 24/04/2020)'; 'Amount:' with an empty field; and 'Reason for Valuation:' with an empty field and a lightbulb icon. At the bottom right of the form are two buttons: 'Save' and 'Reset'.

Features of a Form:

Default values	<p>A field can include a default value, <i>Accession Lot</i> in the example above.</p> <p>A default value is typically provided by the Form's designer if the value is most often used in a field. Unless it has also been configured to be <i>read-only</i>, you can change it if required.</p>
Date fields	<p>Date fields display a hint indicating the format of dates. When the cursor enters the field, the hint disappears and a calendar pops-up:</p>  <p>Either:</p> <ul style="list-style-type: none"> » Select a date in the current month. » Use the previous / next arrows in the pop-up header to select a past / future date. » Key a date in the required format.
Attachment fields	<p>An attachment field allows users to search for and link a record to the current record (the record being created / edited with the Form).</p> <p><i>Valued by</i> is an attachment field linking to the Parties module. As characters are entered in an attachment field, matching records are dynamically listed. Select the record to attach:</p> 



<p>Optional, mandatory and recommended fields</p>	<p>Fields are configured to be optional (if you have data, enter it in the field), mandatory or recommended.</p> <p>Fields that are mandatory or recommended contain an icon:</p> <p> The field is mandatory (<i>Valued By</i> in the first image above). It is not possible to save a record if a value has not been provided for a mandatory field. Feedback will be provided indicating that the field must be completed.</p> <p>Place the cursor over the icon for a definition of the icon:</p>  <p>Or click the icon to display a message about what is required of the field.</p> <p> Completing the field is recommended but not required (<i>Reason for Valuation</i> in the first image above). The Form will save whether there is a value in the field or not.</p> <p>Place the cursor over the icon for a definition of the icon:</p>  <p>Or click the icon to display a message about what is required of the field.</p>
 and 	<p>These buttons are disabled (grey) until data is entered in a field. An enabled button is coloured:</p> 
 and 	<p>Collapse and expand sections.</p>

1. Enter data and click  to save the data and clear the Form ready for a new record.


When the Form is saved, it undergoes standard EMu validation (checking that numeric values are entered in fields configured as numeric, for instance; checking that mandatory fields have been completed). A record that fails validation will need to be corrected before it can be saved.

Feedback is provided indicating how one or more fields fails validation.

2. When you have finished entering data, click:

	<p>Return to the Home screen.</p> <p>If you belong to more than one project, the Home screen lists all the projects you belong to. If you belong to one project only, the Home screen lists the data entry Forms and any Editors you have access to.</p>
	<p>Return to the list of Forms and Editors you have access to in the current project.</p>

-OR-

Click  to exit the application.



Note: It is possible to bookmark a Form in order to return directly to it rather than navigating through projects.

Working with images and other multimedia


A Sapphire Form can include an image control that displays one or more images attached to the record you are adding/ editing. The image control may also enable you to manipulate an image (add polygons to it), remove it from the current record, and upload an image and attach it to the current record.




Note: Some Forms will allow you to upload any type of multimedia, not just images.

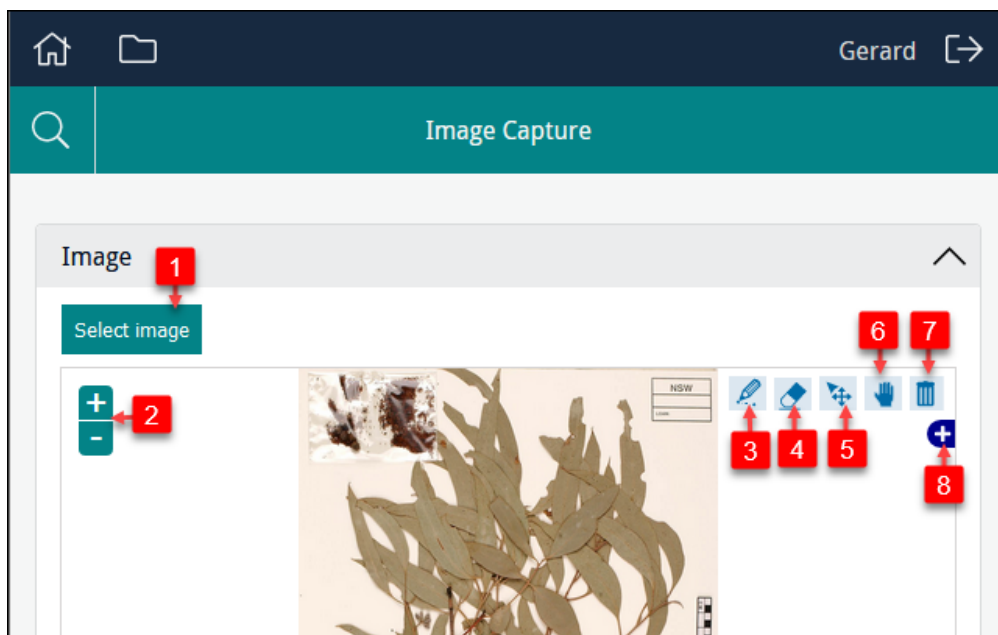
If you add an image to the current record (a Catalogue record for example) by uploading a new image, several things will happen when the Form is saved:








- » A new record for the image will be created in the Multimedia module.
- » The new Multimedia record will be attached to the current record.

It is important to keep in mind that a Form is typically designed to edit records or create new records in modules such as the Catalogue or Parties: the purpose of an image control is to allow you to view, add or remove images attached to the current record. While you create Multimedia records when you upload a new image, when you use the **Remove**  button you are removing an attachment to the Multimedia record, not deleting a record from the Multimedia module.



If the search icon  displays in a Form, it is possible to search the Multimedia repository for an image and display it (see *How to search for and edit a record* on page 20 for details):



#	Icon	Label	Description
1		Select Image	See <i>Select Image</i> (page 14) for details.
2		Zoom in / out	Click to zoom in / out an image. Alternatively, use your mouse scroll wheel to zoom in and out.
3		Draw Polygons	See <i>Polygons</i> (page 15) for details.
4		Remove Polygons	See <i>Polygons</i> (page 15) for details.
5		Edit Polygons	See <i>Polygons</i> (page 15) for details.
6		Navigation	Drag an image around the display area of the Form.
7		Remove an attached image	See <i>Attached images</i> (page 17) for details.
8		View attached images	See <i>Attached images</i> (page 17) for details. This icon will only display if more than one image is attached to the current record.

Uploading multimedia files


A Sapphire Form can display an image already saved in your system and / or allow you to upload an image. Some Forms will allow you to upload any type of multimedia, not just images.

There are several methods for uploading a multimedia file:

- » Depending on the multimedia control added to the Form, you will be able to:
 - » Click **Select Image** (only for image uploads) or **Browse** to locate and select a file that is accessible to your computer / device.
 - » Use your device's camera to capture a new image and upload it with the Form.

-OR-

- » Drag and drop a file from your computer / device directly onto the multimedia control on the Form.


If the Form includes an image control and already displays an image, the new image will be accessible using the **View attached images**  icon. See *Attaching and viewing images* (page 17) for more details.




Note: If an upload times-out, it is likely that the file size of the upload is too large. It will be necessary to reduce the number of files being uploaded or, if possible, compress the multimedia file(s).

Polygons

It is possible to draw polygons on images to highlight certain areas of the image. To draw a polygon on an image:

1. Click .
2. Click somewhere on the image, move the cursor to another point on the image and click again.

 **Note:** Be sure to release the mouse button, otherwise you will drag the image.

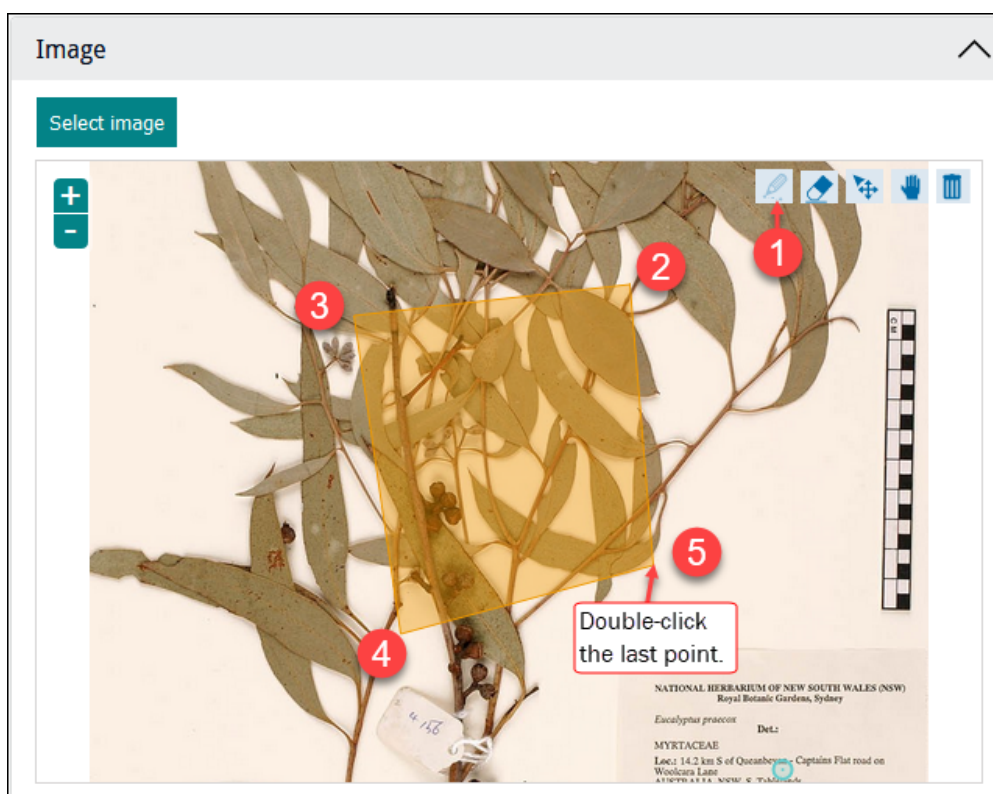
A line is drawn between the two points.



3. Move the cursor to another point on the image and click again.

A three sided polygon is created.


Move the cursor and click additional points to add sides to the polygon as required.

4. Double-click the final point to complete the polygon:



 **Note:** Until you select another icon, the Draw tool  will remain active.

It is possible to adjust (resize, rotate) and move a polygon:

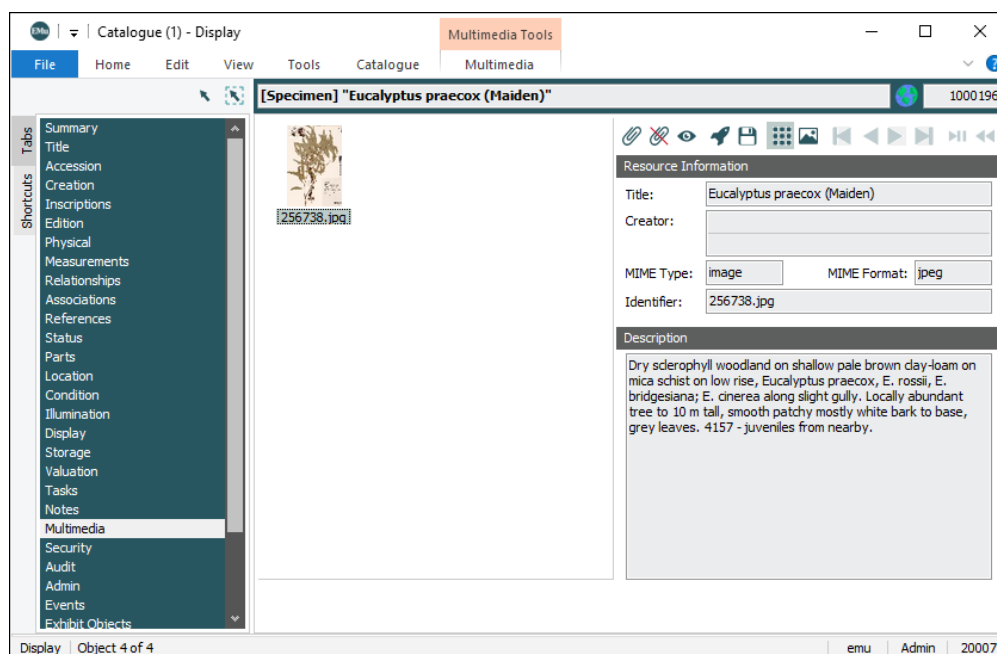
1. Click .
2. Click a polygon.

Each polygon has two edit points: (1) the centroid (the central point of the polygon) and (2) a point outside the polygon:

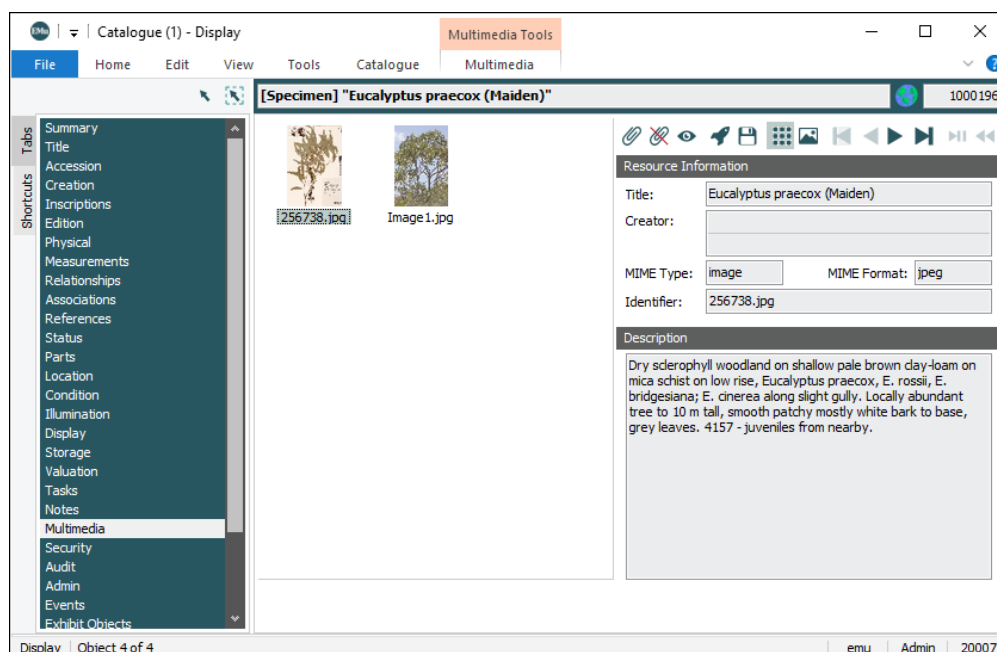
Attaching and viewing images


A Form can include an image control to display an image already loaded into your system and attached to the current record. The image control may also enable you to upload / add an image and attach it to the current record.

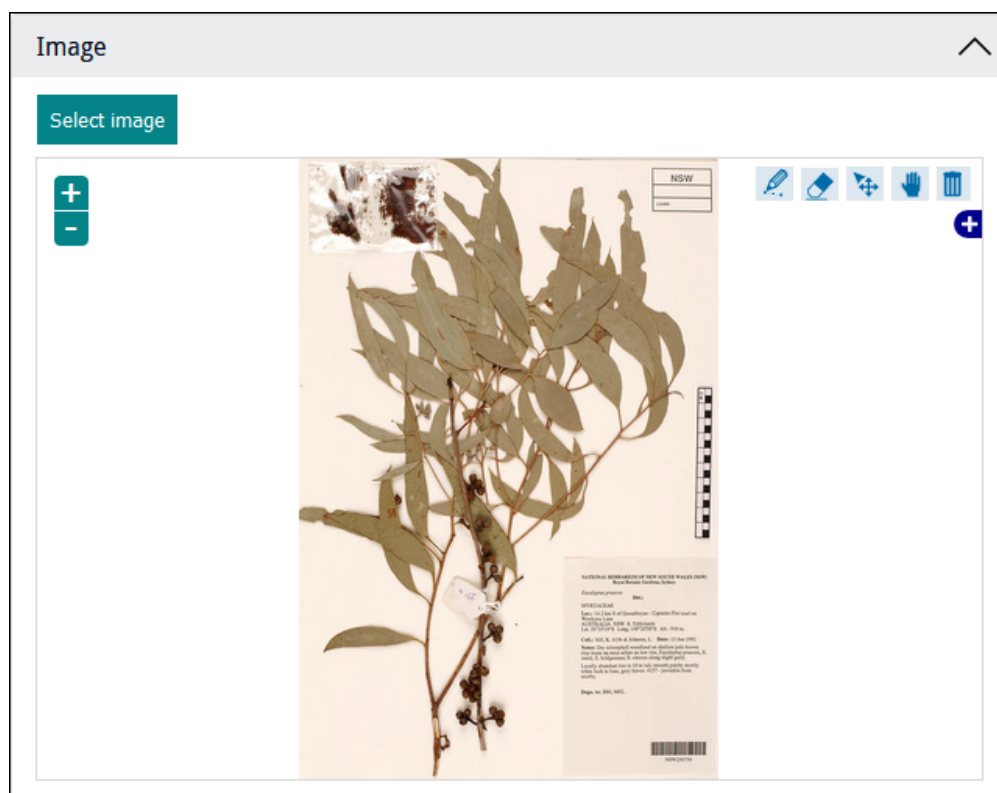
A record, in the Catalogue module for example, can be attached to a single image record in the Multimedia module:



or to multiple image records:



The image control in a Sapphire Form can display the attached images. If only one image is attached, the **View attached images**  button will not display. In this example, the Form pulls details from the Catalogue record with two Multimedia records attached:



Depending on the type of image control included on a Form, it may be possible to:

- » Upload an image and attach it to the current record.
- » Remove an image from the current record.


To add a new image, either:

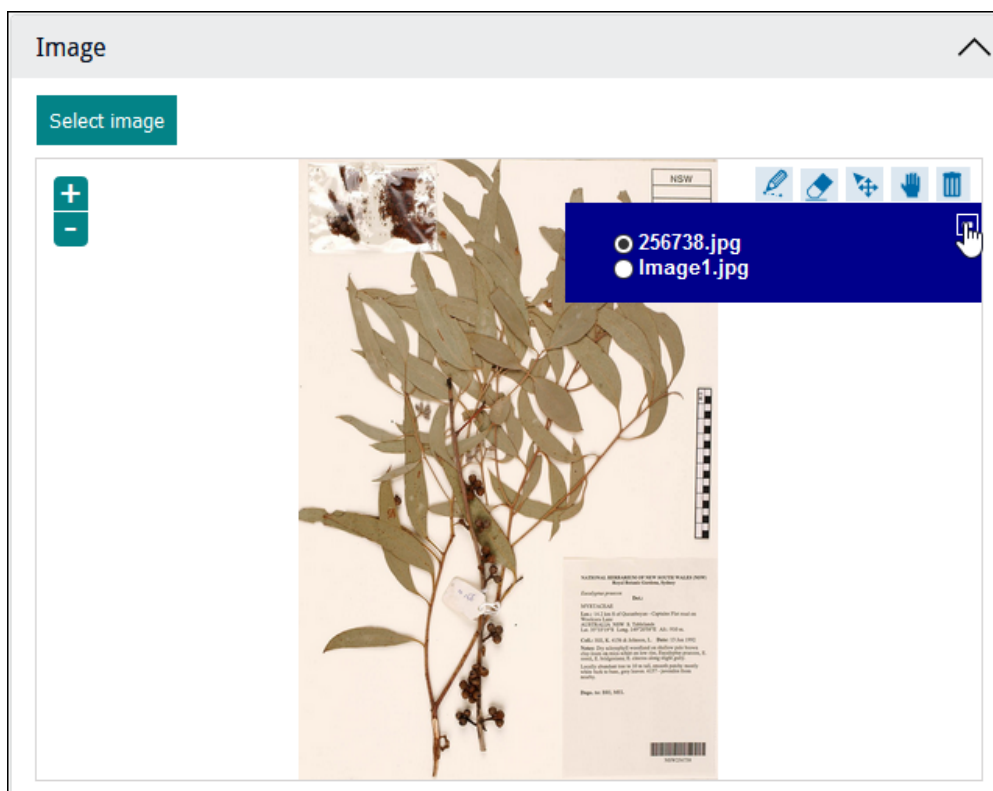
- » Click **Select Image**, and locate and select an image.
- OR-
- » Drag and drop an image from your computer / device to the Form.
- OR-
- » If enabled, capture an image with your device's camera.

When the Form is saved:

- » A new record for the image will be created in the Multimedia module.
- » The new Multimedia record will be attached to the current record.


To view other attached images:

1. Click the  on the right side of the Form:



2. Select the radio button beside the image you wish to view.

To remove an image from the current record:

1. View the image.
2. Click .

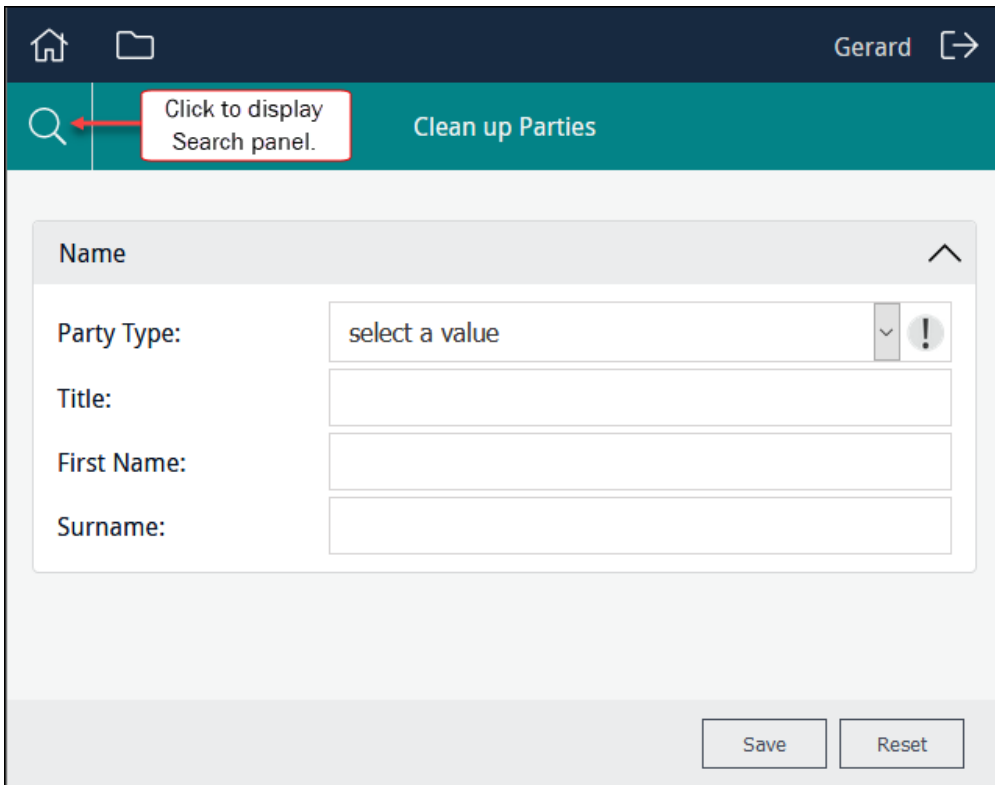


Note: This action deletes the attachment between the current record (a record in the Catalogue module for instance) and the record for the image in the Multimedia module. It does not delete the Multimedia record, and the image could be reattached to the record or attached to another record.

How to search for and edit a record

A search option can be included in a Form, enabling users to search a module for a record and edit it.

This Form could be used to create a new record by completing the *Name* fields and clicking **Save**:



The screenshot shows a web interface for a form titled "Clean up Parties". At the top left, there are navigation icons for home and folders. The user's name "Gerard" and a share icon are at the top right. A search icon is located on the left side of the form header. A red callout box with a white background and a red border points to the search icon, containing the text "Click to display Search panel.". The form itself is titled "Name" and contains four input fields: "Party Type:" with a dropdown menu showing "select a value", "Title:", "First Name:", and "Surname:". At the bottom right of the form, there are two buttons: "Save" and "Reset".

The Form could also be used to search for a record and edit it:

1. Click the **Search**  icon.

A search panel slides out from the left.

2. Enter your search term and click **Go**.

Search results are displayed as a list below the Search box:

Home Folder Gerard →

Search

sally smith Go

1 records found.

Smith, Mr Sally

New record

3. Click the record you wish to edit.
The Form re-displays and the fields are populated with values from the selected record.
4. Make your changes (changing the person's title in this case) and save the record:

Home Folder Gerard →

Search Clean up Parties

Name

Party Type: Person

Title: Ms

First Name: Sally

Surname: Smith

Save Reset

How to use a Sapphire Editor to clean up data and process attachments

A Sapphire Editor is, for all intents and purposes, a Global Replace tool. It enables authorised users to search for and select one or more records and:

- » Update a value in a field (e.g. perhaps replacing variations of a value with an approved value).
- » Attach the selected records to another record.

Example use case

Sapphire facilitates the rapid capture of data. In a typical scenario, Data Entry users key data from hard copy documents directly into a data entry Form, creating a record in an EMu module.

While this is an efficient way to enter data and create records, it is likely that data captured from documents produced by different writers over time will include inconsistencies and errors. Sapphire is designed to facilitate data cleansing. Editors enable authorised users to update the values in fields, perhaps replacing variations of a value with an approved value.

An Editor can perform far more complex tasks however.

Data captured by Data Entry users can include information that in EMu would be stored as records in various modules linked together. For example, data captured from a specimen label might include details of the specimen, the collector and a location in which the specimen was collected. In EMu this information might be recorded as:

- » A Catalogue record with details of the specimen.
- » A Parties record with details of the collector.
- » A Collection Events record with details of the location, etc.

Both the Parties and Collection Events records might attach to the specimen's Catalogue record.

While it is possible to create data entry Forms to capture all of this information as separate records and create attachments between them, it may be far more efficient to separate these processes and have Data Entry users rapidly capture the data (as text) and then process the captured data to create attachments using an Editor.

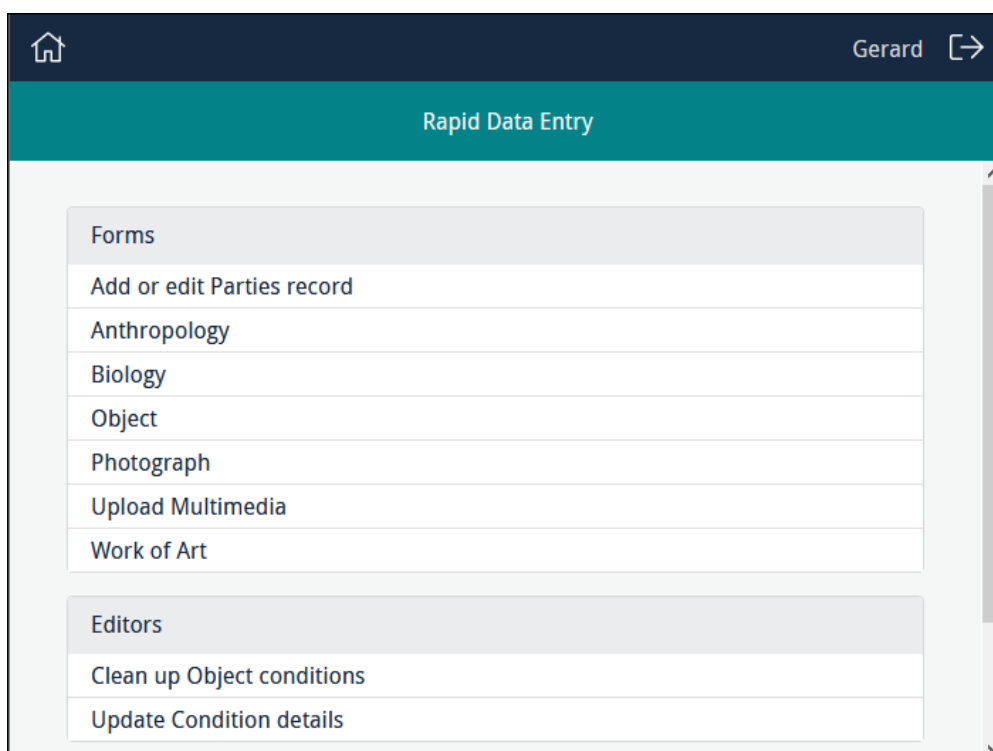
For example, during data entry a collector's name may have been captured from specimen labels and stored in a text field in a Catalogue record for the specimen. A Sapphire Editor can then be used to search for and select Catalogue records for specimens collected by the same collector and create an attachment from the selected records to a Parties record for the collector.

Details for Programme Leads about how to configure a Sapphire Editor can be found in *How to create and edit an Editor* (page 98).

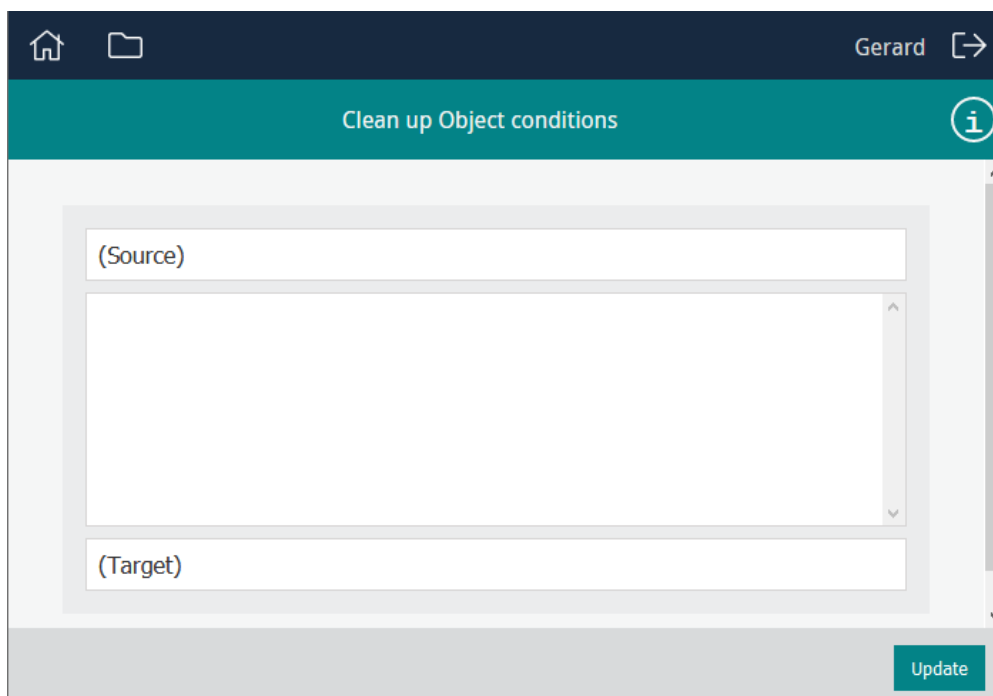
How to use an Editor

For a user, the steps for creating attachments and updating values in a field are similar:

1. Select an Editor from the Editors list:

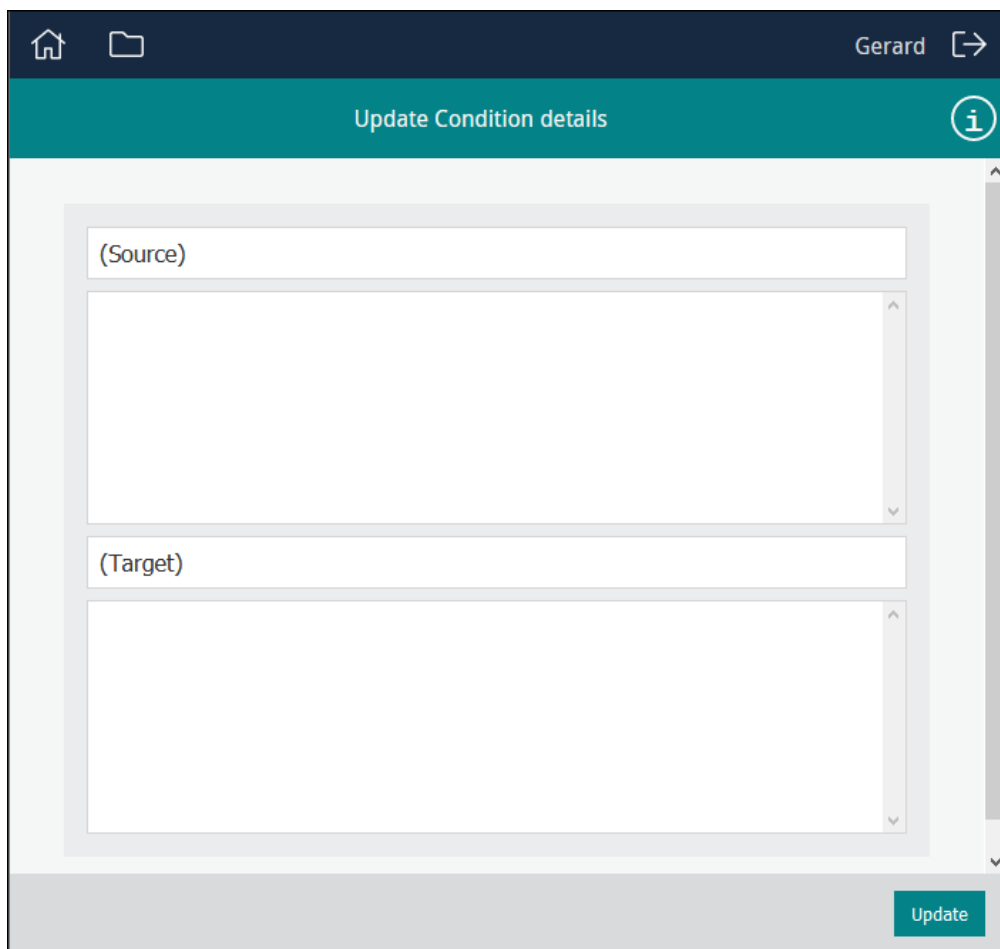


With this Editor you search for and select one or more records and update a value in a field:



Sapphire Guide

With this Editor you search for and select or more records, and attach them to a record selected in the *Target* pane:



The screenshot shows a software interface titled "Update Condition details". At the top, there is a dark blue header with a home icon, a folder icon, the name "Gerard", and a right-pointing arrow icon. Below the header is a teal bar with the title "Update Condition details" and an information icon (a lowercase 'i' in a circle). The main content area is a light gray frame containing two sections: "(Source)" and "(Target)". Each section has a large, empty white rectangular area with a vertical scrollbar on the right side. At the bottom right of the frame, there is a teal button labeled "Update".

(Source)	<p>Here we search for records:</p> <ol style="list-style-type: none"> 1. Enter a search term. 2. Hit <code>ENTER</code>. <p>A list of matching records will display in the <i>Source</i> pane.</p> <p>A Form to update data</p> <p>If the Editor is designed to update values in a field, we might search <i>Source</i> for all variations of a term.</p> <p>For example, a field might hold a Condition description and values captured by Data Entry staff might include a wide variety of terms and typos, e.g.:</p> <ul style="list-style-type: none"> » Good » Fair » Excellent » Very Good » V Good » V Gd » Godo » Poor » Bad » Not good <p>We would search for these terms and select one or more records we wish to update with a value selected in the <i>Target</i> field.</p> <p>A Form to create attachments</p> <p>If the Editor is designed to create attachments, we search <i>Source</i> for one or more records to attach to a record we will select in the <i>Target</i> pane.</p> <p>For example, during a digitisation project, a data entry Form was used to create Catalogue records for specimens. The name of a specimen's collector was captured from specimen labels and stored in a text field (e.g. <i>Notes</i>) in the specimen's Catalogue record.</p> <p>An Editor can allow us to enter a collector's name in <i>Source</i> and perform a search for it in the <i>Notes</i> field; any matching records will be listed in the <i>Source</i> pane. In this example, we would select all Catalogue records we want to attach to a Parties record for the collector.</p>
----------	---

(Target)

Here we enter a search term to search EMu for:

» An approved value

-OR-

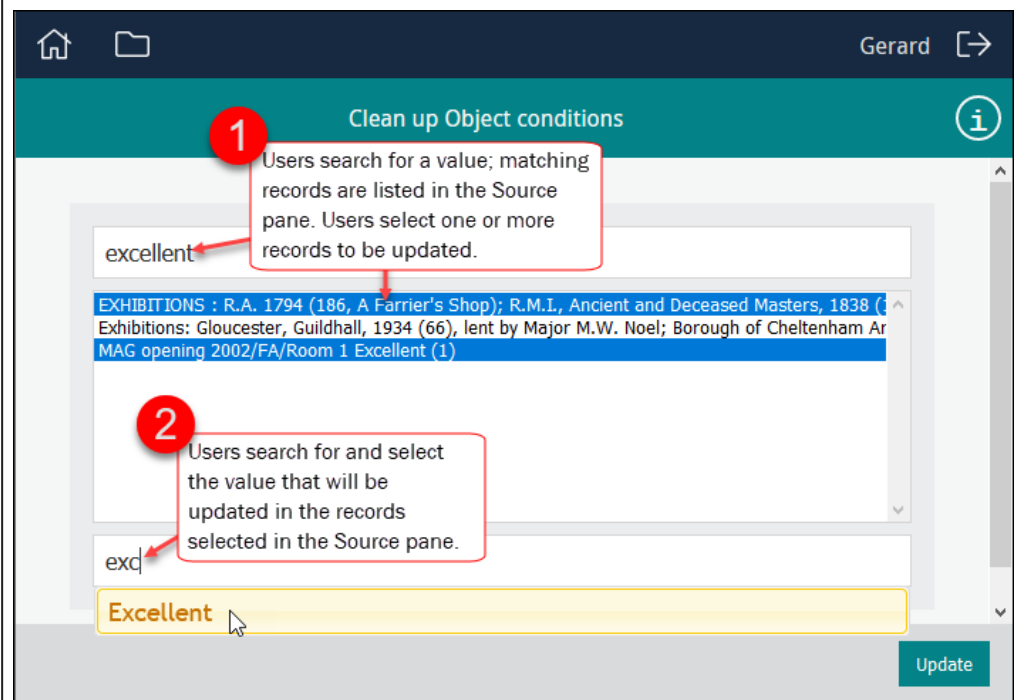
» An EMu record to attach to records selected in the *Source* pane.

One entry can be selected.

A Form to update data

If the Editor is designed to update values in a particular field (*Condition Status* in this example), *Target* will hold values from a Lookup List:

1. Start keying the required term.
A list of filtered terms will display.
2. Select the term to be used:



In this example, when the Form is updated, the *Condition Status* of the records selected in the *Source* pane will be updated to *Excellent*.

A Form to create attachments

If the Editor is designed to create attachments, the *Target* pane will return records from the appropriate module (*Parties* in this example):

1. Enter search details and hit `ENTER`.
2. Select the record to use to create attachments:

Attachments will be created from the records selected in the *Source* pane to the record selected in the *Target* pane.

In this case, an attachment would be created from the *Checked By: (Condition Check)* field in the Catalogue records to the Parties record for the person who performed the condition check.

See *Example: creating attachments* (page 28) for more details.

2. Click **Update**.

If all goes well, a message indicating the success of the update will display.

Example: creating attachments

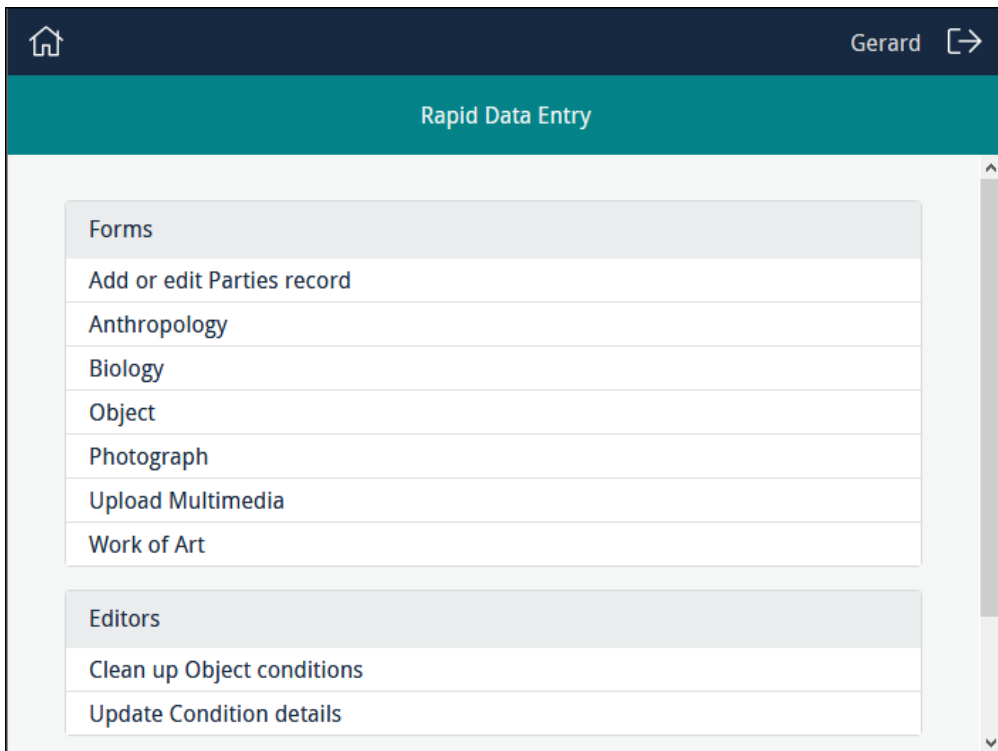
In this example details about object conditions, including who performed a condition check, were keyed into the *Notes* field of records in the Catalogue module. In EMu, details about the person performing a condition check are recorded in the Catalogue record as an attachment to a Parties record.

Using the **Update Condition Details** Editor in this example, it is possible to:

1. Search the *Notes* field for details about the person who performed condition checks, and select all matching Catalogue records.
2. Identify the record in the Parties module for the person who performed the condition check.
3. Update the selected Catalogue records by creating an attachment from the appropriate field in the Catalogue record, *Checked By: (Condition Check)*, to the person's Parties record.

To create attachments:

1. Select an Editor from the Editors list:



In this case we select **Update Condition details**:

2. In the *Source* field, enter a search term and press `ENTER`.

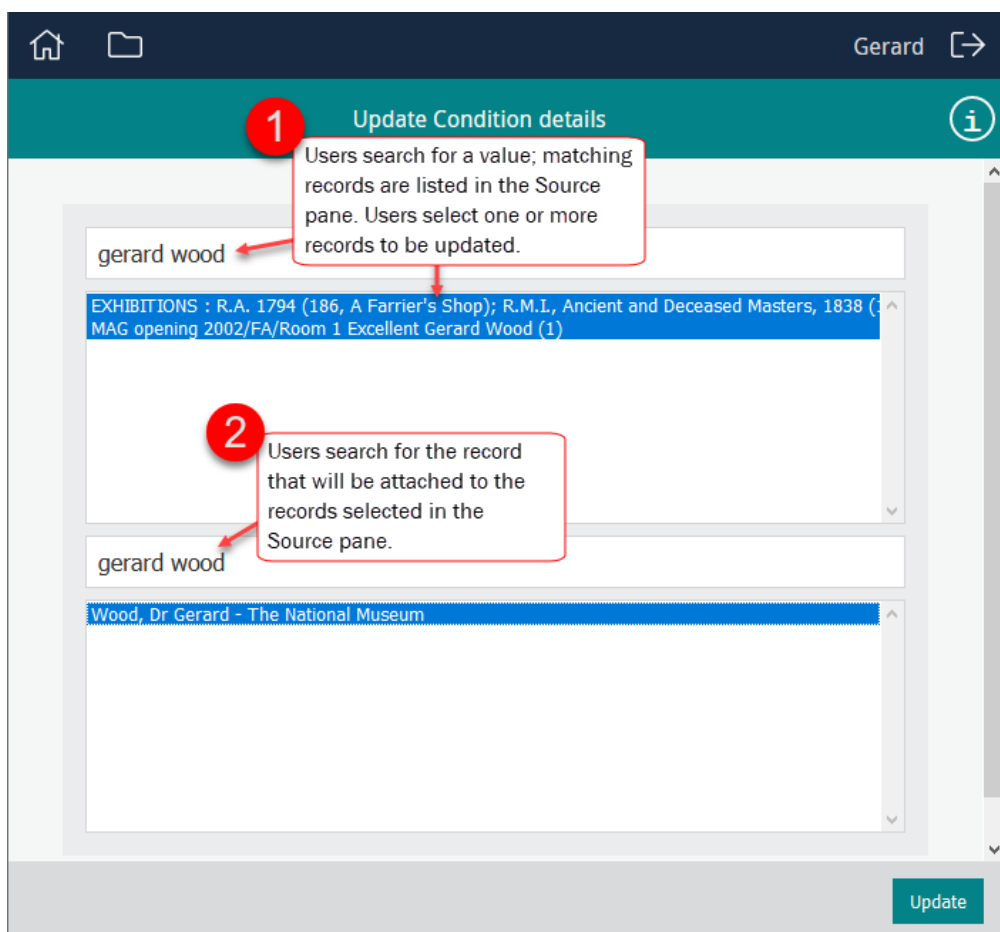


Note: A search in a Sapphire Form obeys the same rules as a search in an EMu module.

In this example the Editor has been configured so that a search in *Source* will search the *Notes* field in the Catalogue module. We search for names of people who performed a condition check.

Matching records are listed in the *Source* pane.

3. In the *Source* pane select one or more records.
4. In the *Target* field, search for the record to attach to.
A search is performed and details are returned from the relevant EMu module.
5. Select the record which will be used to create attachments to the records selected in the *Source* pane:



6. Click **Update**.

An attachment is created from each selected record in the *Source* pane to the record selected in the *Target* pane.

In this example, an attachment would be created from the *Checked By: (Condition Check)* field in the Catalogue records to the Parties record for the person who performed the condition check:

File Home Edit View Tools Catalogue

New Record Cancel Previous New Additional Sort Reports Check Spelling All Records Clear Current Record Add Record Invert

File Search View Tools Select

[Work of Art] "Ludlow Castle from Whitcliffe, Shropshire" [1928.118], Ibbetson, Julius Caesar 18

Condition Check

Condition Status: Excellent Date Checked: 04/05/2020

Checked By: Wood, Dr Gerard - The National Museum

Condition Details:

Handling:

Next Due

Notification Date: Condition Period: Years Months

Condition Check History

Date Ch...	Checked By	Condition Status
1 04/05/2...		Excellent
2 04/05/2...	Wood, Dr Gerard - The National Museum	Excellent

Display | Object 3 of 3 emu Admin 2007

Programme Lead

Programme Leads create and edit Forms from within the project Dashboard. It is the Programme Lead's responsibility to:

- » Create projects, assign access permissions and make the project available to users (page 39).
- » Create and maintain data entry Forms and Editors (page 44).

A data entry Form can be designed to:

- » Capture or edit one record at a time (page 33)
- » Perform some special processing (page 36)

A Form to capture or edit one record at a time

Users enter data in a Form which creates or updates one record in an EMu module as the Form is saved: there is a one to one relationship between the data entered in the Form and a record in an EMu module.

The following screenshot shows the design of a data entry Form to capture details of a `Person Party`. The Form comprises a single section with four fields (only the first is shown) and a search section:

🏠
Gerard ↔

Title:

Description:

Identifier: Table:

Processing script: Deferrable:

✓ Permissions ^

Everyone:

+ Add permission

📄 Sections ^

Section ✕ ^

Label: ID:

Field ✕ ^

Identifier: Column:

Label:

Kind: Type:

Access: Requirement:

Lookup: Level:

Default:

Ref. Table: Ref. Column:

Match Limit: Image Width (Best Fit):

Filters

+ Add filter term

Title ✕ ▾

First Name ✕ ▾

Surname ✕ ▾

+ Add Field

+ Add section

🔍 Search ▾

📄 Custom Layout ^

+ Add layout

Save
Reset

When accessed by a user, the Form will display as:

When details are entered and the Form is saved, a record is created in the Parties module.

The search option can be used to locate a record in the Parties module, which can be edited and saved.

Some features of this Form:

#	Feature
1	Clicking the information icon will display a pop-up with details about the purpose of the Form: <div data-bbox="258 1144 1264 1417" style="border: 1px solid black; padding: 10px; margin-top: 10px;"> <p>Description Use this form to add a Parties record or search for and edit an existing Parties record Hide details</p> <p style="text-align: right;">Close</p> </div>
2	This field has been configured as mandatory. The Form cannot be saved unless a value is entered in the field.
3	A drop list of available values for the field.
4	Searches the Parties module for a record and displays its values in the Form, where they can be edited and saved back to the Parties module.

Perform some special processing

Rather than creating or editing a single record when a Form is saved, the Form data can be processed by a script to manipulate it in any number of ways.

A typical use of a Processing Script is bulk creation of records.



Note: Processing scripts are written in perl, typically by Axiell developers. A number of scripts are available and can be provided on request; additional scripts can be written to meet a customer's requirements. Please contact Axiell Support to discuss.

Each project has a directory on the IMu server located beneath the `forms` directory. Processing scripts for a project are saved to this directory.

It is not necessary to enter the name of the script in the *Processing script* field if there is only one script assigned to the Form and the script is named using the Form's Identifier with a `.pl` suffix (this becomes the default script), e.g.:

`imaging.pl`

If there is more than one processing script available (and you do not wish to use the default script), enter the name of the script to use in the *Processing script* field:

The screenshot shows a web form configuration interface. At the top, there are navigation icons (home and folder) and the user name 'Gerard' with a refresh icon. The form contains several fields:

- Title:** Add or edit Parties record
- Description:** Use this form to add a Parties record or edit an existing Parties record
- Identifier:** form1259
- Table:** eparties
- Processing script:** (This field is highlighted with a red border and is currently empty.)
- Deferrable:** No (with a dropdown arrow)

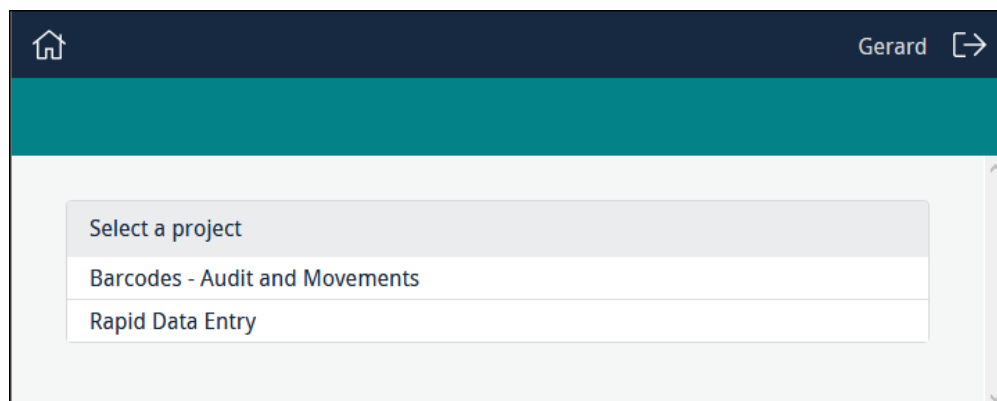
Below these fields is a section titled 'Permissions' with a checkmark and an upward arrow. It contains:

- Everyone:** use (with a dropdown arrow)
- + Add permission** (with a plus icon)

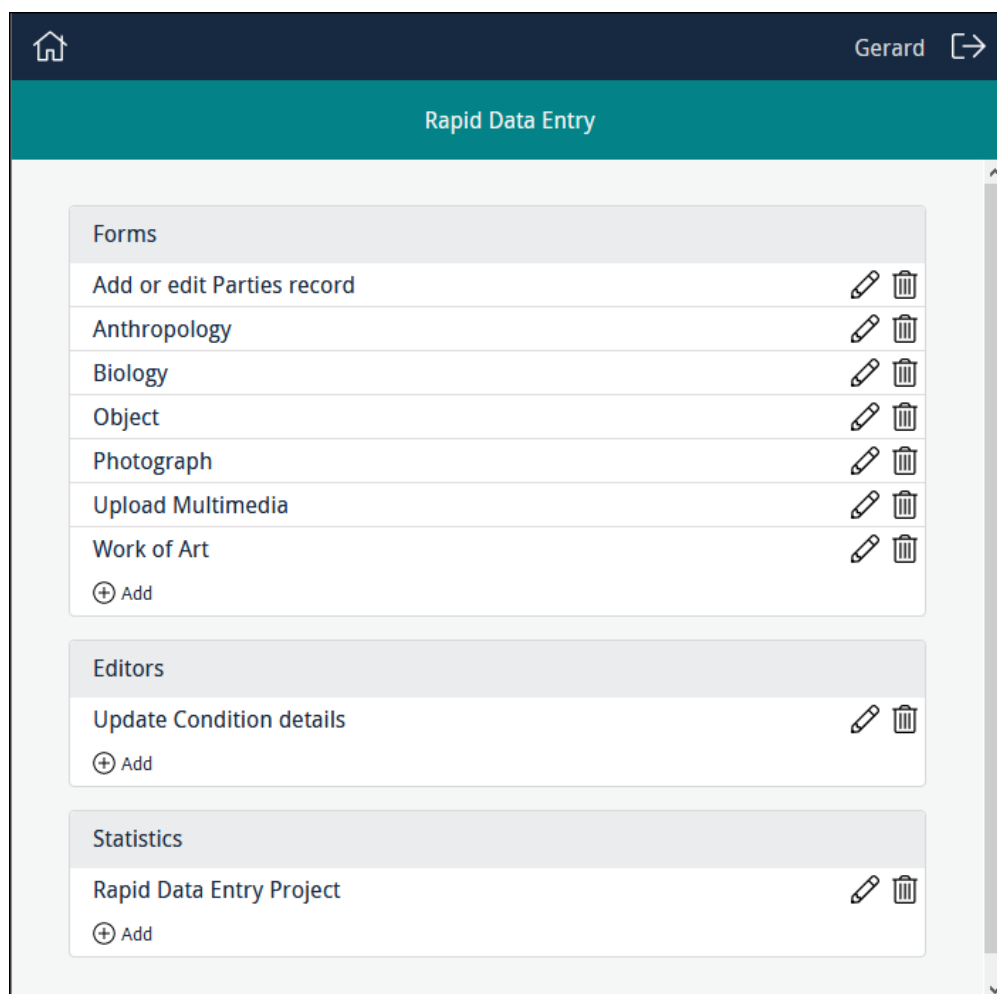
At the bottom right of the form, there are two buttons: 'Save' and 'Reset'.

Logging in a Programme Lead

If you are a member of more than one project, you will be presented with a list of the projects to which you belong when you log in. Select the project you wish to work with:






If you are a member of only one project, or you selected a project from the list of projects available to you, any Forms, Editors and Statistics tools already created for the project will be listed, along with options to edit, delete and add new instances of them:




Selecting a Form, Editor or Statistics tool title (e.g. **Add or edit Parties record** in the example above) will display the project element as it does for your users (ready for use).


As well as using a Form, Editor or Statistics tool, a Programme Lead has additional options:

	<p>Edit the Form, Editor or Statistics tool.</p> <p>For details, see:</p> <ul style="list-style-type: none"> » <i>How to create a data entry Form</i> (page 44) » <i>How to create and edit an Editor</i> (page 98) » <i>How to display Statistics</i> (page 108)
	<p>Delete a Form, Editor or Statistics tool.</p> <p>A message will display asking that you confirm the deletion.</p>
	<p>Create a new Form, Editor or Statistics tool.</p> <p>For details, see:</p> <ul style="list-style-type: none"> » <i>How to create and edit a data entry Form</i> (page 44) » <i>How to create and edit an Editor</i> (page 98) » <i>How to display Statistics</i> (page 108)

When you have finished managing your project:

1. Click  to return to the list of projects you have access to (if you are a member of more than one project) or the list of project elements (Forms, Editors, etc.) you have access to (if you are a member of only one project).

-OR-

- Click  to exit the application.

How to create a project and make it available to users

Creating a project and making it available to users is the responsibility of the Programme Lead. It involves three steps:

1. Name the project in the EMu Registry.
See *Name the project* (page 39) for details.
2. Define user access in the EMu Registry:
 - » Assign permissions to users to determine what elements (Forms, Editors, Statistics) of the project they have access to (page 41).
3. Create Forms (page 44) and Editors (page 98), and Statistical graphs (page 108).



Note: As with assigning permissions in EMu, it is good practice to assign permissions to a group and to make users members of the group. Permissions can be assigned to users, of course, and these will take precedence over group level permissions. See the [EMu Help](#) for details about the order in which Registry entries are assigned.

- » Define a user's role in a project (page 42).
- » Define which projects the user has access to (page 43).

Registry entries

The following Registry entries are required to give a user access to Sapphire. Once set, authorised users are able to commence working with a Form or, in the case of a Programme Lead, create and manage data entry Forms, Editors and Statistical graphs.

Name the project

Name the project and define a unique project ID.

For this we use the following Registry entry:

```
User | user | Project | id | Title | project title
Group | group | Project | id | Title | project title
Group | Default | Project | id | Title | project title
```

where:

- id* is a unique ID for the project.
It is recommended that *id* is constructed from letters, numbers, hyphens and underscores only.
id is referenced by other Registry entries that define who has access to what elements of the project.
- project title* The name of the project as it will display to users when they log in to Sapphire.
If the user belongs to more than one project, a list of project titles will be presented.



Note: Typically a Group | Default setting is used.

Example

The following example provides a name and unique id for a Museum's Herbarium Digitisation project:

Field	Value
Key 1	Group
Key 2	Default
Key 3	Project
Key 4	botany-herbarium
Key 5	Title
Value	Herbarium Imaging

Determine who has access to what for each project

Create a Dashboard with the project elements that users will have access to when they access a project.

For this we use the following Registry entry:

```
User | user | Project | id | Dashboard | element;element;...
Group | group | Project | id | Dashboard | element;element;...
Group | Default | Project | id | Dashboard | element;element;...
```

where:

id is the unique ID for the project defined when the project was named (page 40).

element;element;... is a semicolon separated list of elements that the user / group will have access to when they access this project.

Values for *element* are:

» Forms

The user / group will have access to the project's Forms.

» Editors

The user / group will have access to the project's Editors.

» Statistics

The user / group will have access to statistical graphs configured for the project.

Project elements are displayed on the project Dashboard in the following order:

» Forms

» Editors

» Statistics

Example

The following example provides user `gerard` with access to all available elements when working with the Herbarium Digitisation project:

Field	Value
Key 1	User
Key 2	gerard
Key 3	Project
Key 4	botany-herbarium
Key 5	Dashboard
Value	Forms;Editors;Statistics

Define a user's role

Define the roles of users and / or groups within the project.

For this we use the following Registry entry:

```
User | user | Project | id | Role | role  
Group | group | Project | id | Role | role  
Group | Default | Project | id | Role | role
```

where:

id is the unique ID for the project defined when the project was named (page 40).

role is the role assigned to a user or group.

Values for *role* are:

» lead

The Programme Lead. The Programme Lead is able to create, configure and manage the project elements.

» member

Anyone other than the Programme Lead.

Example

The following example makes user `gerard` the Programme Lead for the Herbarium Digitisation project:

Field	Value
Key 1	User
Key 2	gerard
Key 3	Project
Key 4	botany-herbarium
Key 5	Role
Value	lead

Define which projects a user has access to

Specify which projects a user / group has access to and the order in which they will be listed when a user logs in to the project Dashboard.

For this we use the following Registry entry:

```
User | user | Project | id;id;...
```

```
Group | group | Project | id;id;...
```

```
Group | Default | Project | id;id;...
```

where:

id;id;... is a semicolon separated list of project IDs defined when the project was named (page 40).

The order in which the IDs are listed is the order in which they will be listed when a user logs in to the project Dashboard.

Example

The following example specifies that user `gerard` has access to two projects and the Herbarium Digitisation project will be listed before the Palaeontology Imaging project:

Field	Value
Key 1	User
Key 2	gerard
Key 3	Projects
Value	botany-herbarium;paleo

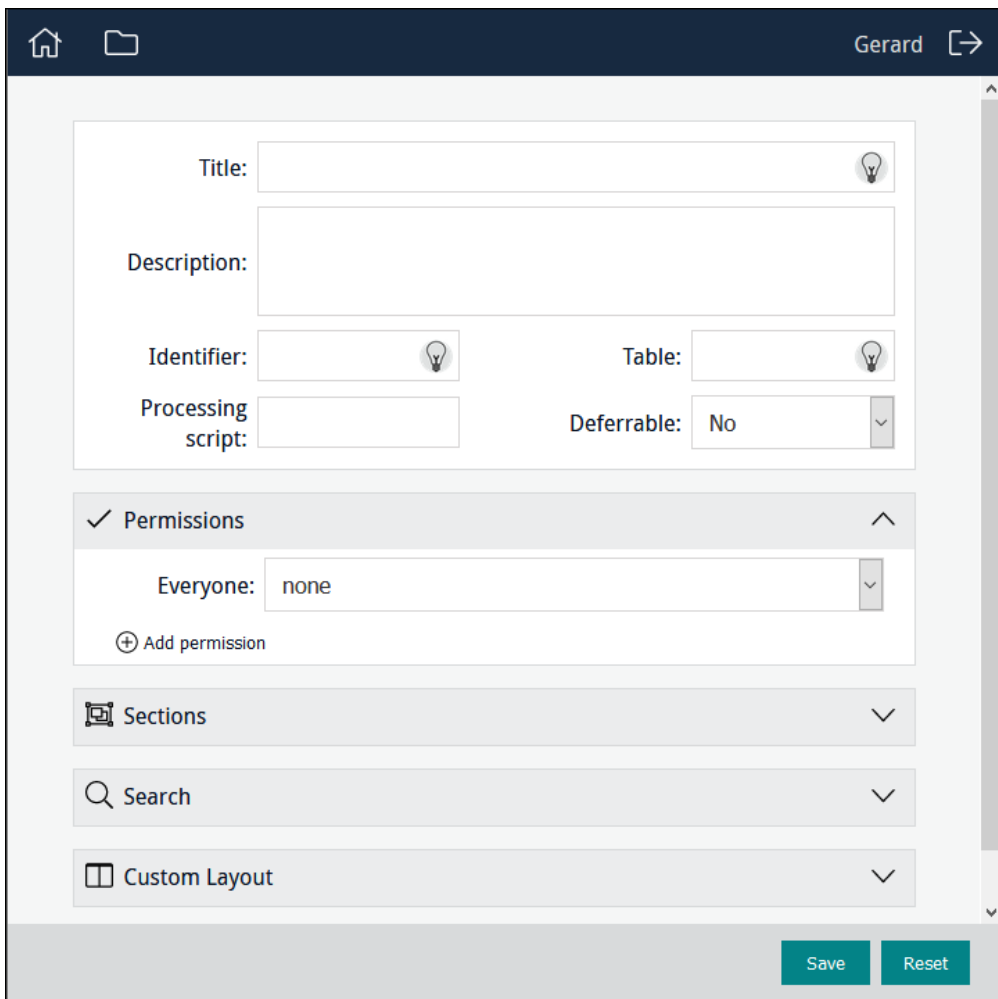
How to create and edit a data entry Form

When a Programme Lead accesses a project, any Forms, Editors and Statistics tools already created will be listed, along with options to edit, delete and add new instances of them.

To create a Form:

1. Click  in the Forms section.

The Form creator displays:


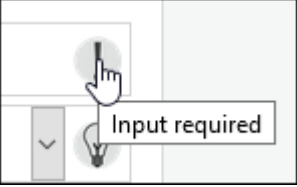




The screenshot shows the Form creator interface. At the top, there is a dark blue header with a home icon, a folder icon, and the name 'Gerard' with a refresh icon. The main content area is light gray and contains several form fields and sections. The 'Title' field is at the top, followed by a 'Description' text area. Below these are 'Identifier' and 'Table' fields, each with a lightbulb icon. The 'Processing script' field is below 'Identifier', and the 'Deferrable' field is below 'Table', with a dropdown menu showing 'No'. Below these fields are four expandable sections: 'Permissions' (checked), 'Sections', 'Search', and 'Custom Layout'. The 'Permissions' section shows 'Everyone: none' and an 'Add permission' button. The 'Sections', 'Search', and 'Custom Layout' sections each have a dropdown arrow. At the bottom right, there are two buttons: 'Save' and 'Reset'.

More details:

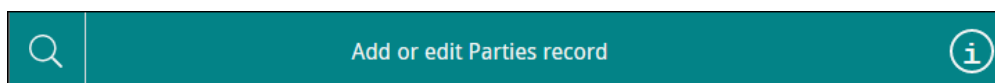
- » To include a Search section in the Form (allowing users to search for a record and update it), see page 91.
- » To customise the layout of a Form, see page 83.

Input fields may contain an icon:

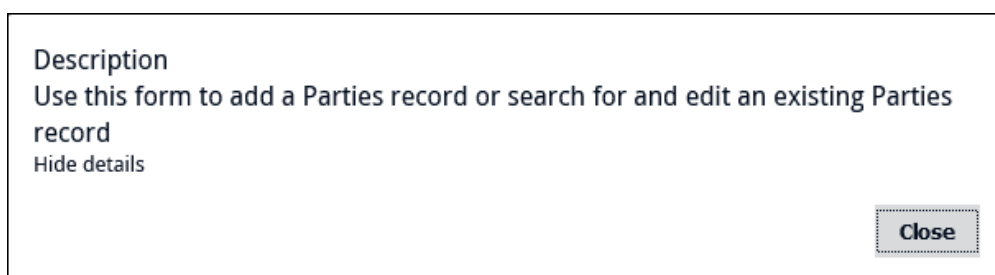
	<p>The field is mandatory. Place the cursor over the icon for a definition of the icon:</p>  <p>Or click the icon to display a message about what is required of the field.</p>
	<p>Completing the field is recommended. Place the cursor over the icon for a definition of the icon:</p>  <p>Or click the icon to display a message about what is required of the field.</p>

- In the *Title* field, give the Form a descriptive name (which will be selected by users who have access to the Form).
- In the *Description* field, add a description of Form or details about how to use it.

When users access the Form, an Information icon will display alongside the title:



Clicking the icon will display a pop-up with the description:




- In the *Identifier* field, add a descriptive identifier (or leave blank and an alphanumeric identifier will be assigned automatically).

For the most part it is irrelevant whether the identifier is an alphanumeric value provided by the system or a descriptive label provided by the Programme Lead as it is only referenced behind the scenes by various components of Sapphire, or used to construct the URL for the Form.

There are situations when it might be handy to assign a descriptive identifier however, e.g.:

- » To substitute an otherwise meaningless alphanumeric value in the URL with a user-friendly descriptor.


» If using a processing script as the script is named using the identifier. See *Perform some special processing* (page 36) for more details.

 **Note:** If providing an identifier, it is recommended that it is constructed from letters, numbers, hyphens and underscores only (no spaces).

- In the *Table* field, specify which EMu module will be updated when the Form is saved (a record will be created or updated in this module).

A filtered drop-list of module names will display as you enter characters in the *Table* field. Select a module name from the list:



 **Note:** When presented with a drop-list of values, it is best practice to select an entry from the list rather than keying the value into the field.

- The *Deferrable* option is intended for wireless devices and affects the order in which records are processed in the event that wireless connectivity is lost.

Options include:

No	This is the default setting and in most cases can be left as such.
Yes	When <i>Yes</i> is selected and wireless connectivity is lost, a queue is created for any new records. Records are added to the end of the queue to ensure that when a connection is re-established the records are saved in the order in which they were created. See <i>The Deferrable system</i> (page 82) for details.

- In the Permissions section, specify permissions for Everyone, or for one or more users or groups. See *How to specify permissions* (page 89) for details.

The screenshot shows the configuration page for a form titled "Add or edit Parties records". The form description is "Use this form to add a Parties record or search for and edit an existing Parties record." The configuration includes fields for "Identifier" (with a lightbulb icon), "Table" (set to "eparties"), "Processing script", and "Deferrable" (set to "No"). A red callout box points to the Identifier field with the text: "When the form is saved, an identifier will be added automatically; alternatively, you can enter your own identifier." Below these fields are sections for "Permissions" (set to "use"), "Sections" (with an "Add section" button), "Search", and "Custom Layout". "Save" and "Reset" buttons are at the bottom right.


8. Click  to add a section to the Form.




A Form can have multiple sections (or just one) with logical groupings of fields.

9. Add a descriptive name for the section in the *Label* field:

The screenshot shows the "Sections" configuration panel. It contains a "Section" entry with a "Label" field set to "Name" and an "ID" field set to "1". There are "Add Field" and "Add section" buttons at the bottom.

Other features:

	<p>A read-only ID is assigned automatically. Section IDs are used when designing a Custom Layout (page 83).</p>
---	---

	This will remove the section along with any fields in the section.
 and 	Collapse and expand sections.



10. Click  to add a field to the section:

Sections
^

Section
X ^

Label: ID:

Field
X ^

Identifier:  Column: 

Label:

Kind: Type:

Lines:

Access: Requirement:


Lookup: Level:


Default:


Ref. Table: Ref. Column:

Match Limit: Image Width (Best Fit):




Filters



 Add filter term

 Add Field



 Add section

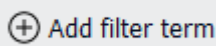
For each field we specify:

<i>Identifier</i>	<p>Used by the system to identify the field.</p> <p>To keep things simple, it is recommended that the <i>Identifier</i> is the same as <i>Label</i> but lower case and without spaces. For instance, if <i>Label</i> is <code>First Name</code>, <i>Identifier</i> could be <code>firstname</code>.</p> <p>An identifier is required, and it must be unique in this Form.</p> <div data-bbox="512 454 1449 533" style="border: 1px solid #ccc; background-color: #f0f0f0; padding: 5px;">  Note: Click a field or use <code>TAB</code> to move between fields. </div>
<i>Column</i>	<p>The back-end name of the field that will be updated when the Form is saved. As the back-end field name is keyed into the <i>Column</i> field, a drop-list of available column names will display. Select a column name from the drop-list.</p> <div data-bbox="512 725 1449 846" style="border: 1px solid #ccc; background-color: #f0f0f0; padding: 5px;">  Note: Unless a processing script allows for it, a column should only appear once on a Form. </div> <p>Fields are restricted to those available in the module specified in the <i>Table</i> field above.</p> <p>Depending on the type of field specified in <i>Column</i>, various values will be auto-completed. For instance, if you specify an attachment field in <i>Column</i>, <i>Type</i> will automatically be set to <code>attachment</code> and <i>Ref. Table</i>, <i>Ref. Column</i> and <i>Match Limit</i> will be completed.</p>
<i>Label</i>	<p>A descriptive label for the field as it will display to the left of the field, e.g. <code>First Name</code>.</p> <div data-bbox="512 1220 1449 1415" style="border: 1px solid #ccc; background-color: #f0f0f0; padding: 5px;">  Note: When adding an image control (page 70) to a Form, it may be better not to include a Field label in order to maximise the size of the image control. This is particularly important when using a custom layout (page 83) with columns. </div>

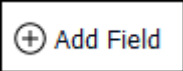
<p><i>Kind</i></p>	<p>Options are:</p> <ul style="list-style-type: none"> » <code>image</code> Use to display an image in the Form. » <code>capture</code> Enable users to locate multimedia of any kind that is accessible to their device and upload it. <div data-bbox="616 465 1449 584" style="border: 1px solid #ccc; padding: 5px; background-color: #f9f9f9;"> <p> Note: <code>capture</code> does not allow users to take a photo with their device and upload it.</p> </div> <ul style="list-style-type: none"> » <code>image-capture</code> Enable users to locate an image accessible to their computer / device and upload it. <div data-bbox="616 741 1449 898" style="border: 1px solid #ccc; padding: 5px; background-color: #f9f9f9;"> <p> Note: <code>image-capture</code> may enable users to take a photo with their device and upload it, depending on their mobile device.</p> </div> <ul style="list-style-type: none"> » <code>mobile-image-capture</code> Enable a user with a mobile device to capture an image with the device's camera and upload it. For details of these four options, see <i>Multimedia fields</i> (page 70). » <code>single</code> The selected value by default. » <code>multiple</code> A table of values, e.g. <i>Other Names: (Person Details)</i> in the Parties module. When a table of values is specified in <i>Column</i>, <i>Kind</i> is automatically set to <code>multiple</code>. See <i>Kind: multiple</i> (page 57) for details.
--------------------	--

<i>Type</i>	<p>Options are:</p> <ul style="list-style-type: none"> » <code>attachment</code> Allow a user to create an attachment. See <i>Attachment field</i> (page 67) for details. » <code>date</code> The date picker pop-up will display when the cursor is placed in the field. See <i>Type: date</i> (page 59) for details. » <code>integer</code> (for integers only). » <code>number</code> (for floating point numbers). » <code>selection</code> Where <i>Column</i> has an associated Lookup List, specifying a <i>Type</i> of <code>selection</code> will present users with a drop-list of read-only values drawn from <i>Column</i>. In this case it is not necessary to specify the name of the Lookup List on <i>Lookup</i>. If a Lookup List is specified in <i>Lookup</i>, users will be presented with values drawn from the specified Lookup List (it can be any Lookup List). See <i>Type: selection</i> (page 61) for details. » <code>text</code> Where <i>Column</i> has an associated Lookup List, specifying a <i>Type</i> of <code>text</code> will present users with a filtered drop-list of read-only values drawn from <i>Column</i>. Filtering occurs as the user enters characters in the field. If a Lookup List is specified in <i>Lookup</i>, users will be presented with values drawn from the specified Lookup List (it can be any Lookup List). See <i>Type: text</i> (page 63) for details.
<i>Lines</i>	<p>Used where:</p> <ul style="list-style-type: none"> » <code>Kind = single</code> -AND- » <code>Type = text</code> <p>Specify the height (number of lines) of the text field.</p>


Access	<p>Determines whether the field can be edited or not. Options are:</p> <ul style="list-style-type: none"> » <code>editable</code> - the default access. » <code>hidden</code> - the field is not shown on the screen. » <code>read-only</code> - the field is visible but not editable. <p>The <code>hidden</code> option is useful when used in conjunction with a default value (see <i>Default</i> below): it is possible to specify a default value for the field and hide the field from the Form's user, preventing the value from being modified and ensuring it is always added to the record created.</p>
Requirement	<p>Options are:</p> <ul style="list-style-type: none"> » <code>optional</code> - the default option; users may enter data in the field or leave it blank. » <code>mandatory</code> - the field must be completed in order for the Form to be saved. » <code>suggested</code> - the field should be completed but it will be possible to save the record if not. <p>An icon is added to fields that are specified as <code>mandatory</code> () and <code>suggested</code> (). If a Form is saved with one or more incomplete <code>mandatory</code> fields, the save will be prevented until the fields are completed.</p> <p>If a Form is saved with one or more incomplete <code>suggested</code> fields, the user will be alerted but will be able to save the record anyway.</p>
Lookup	<p>When a field is specified as having a <i>Type</i> of <code>selection</code> or <code>Text</code> and <i>Column</i> has an associated Lookup List, a user is presented with a list of read-only values drawn from the <i>Column</i> specified.</p> <p>The <i>Lookup</i> option is used:</p> <ul style="list-style-type: none"> » In conjunction with <code>selection</code> or <code>text</code> and <i>Column</i>. Specify the back-end name of a Lookup List to present its values to a user instead of the values drawn from <i>Column</i>. <p>This could be useful, for instance, if <i>Column</i> includes a large number of entries and only a small number are required for the purposes of data entry.</p> <ul style="list-style-type: none"> » Where no <i>Column</i> has been specified when using a Processing Script to perform some special processing (page 36). <p>Specify the back-end name of a Lookup List to present its values to a user. If the Lookup List is a multi-level hierarchy, use the <i>Level</i> field (see below) to specify which level of the hierarchy to display.</p> <p>See <i>Lookup</i> (page 65) for details.</p>

<i>Level</i>	<p>If the Lookup List specified in <i>Lookup</i> is a multi-level hierarchy, use the <i>Level</i> field to specify which level of the hierarchy to display.</p> <p>See <i>Lookup</i> (page 65) for details.</p>
<i>Default</i>	<p>A value to be added to the field by default when a new record is commenced in the Form.</p> <p>Useful in conjunction with the <code>hidden</code> option (see <i>Access</i> above). It is possible to specify a default value for the field and hide the field from the data entry user, preventing the value from being modified and ensuring that it is always added to the record created.</p>
<p>When an attachment field is specified in <i>Column</i>, a user is presented with a text box; when a value is entered in the text box, a search is performed in <i>Ref. Column</i> in the target module (<i>Ref. Table</i>) and matching results are listed (up to the number specified in <i>Match Limit</i>) from which the user can select a record. On saving the Form, an attachment is made from <i>Column</i> to the record selected in the target module (the IRN of the attachment record is saved in <i>Column</i>).</p> <p>When an attachment column is specified in <i>Column</i>:</p> <ul style="list-style-type: none"> » The table to which it attaches is automatically specified in <i>Ref. Table</i>, and <i>Ref. Column</i> is set to <i>SummaryData</i>; a default number (20) is added to <i>Match Limit</i>. » It is possible to specify a Filter to restrict which records are returned by the search of the target module. <p>See <i>Attachment field</i> (page 67) for details.</p>	
<i>Ref. Table</i>	<p>The back-end name of the table to which the attachment field specified in <i>Column</i> attaches. It is auto-completed when an attachment field is specified in <i>Column</i>.</p>
<i>Ref. Column</i>	<p>By default, <i>Ref. Column</i> is set to <i>SummaryData</i>. This is the column searched when a user enters a value in the Search field of the Form.</p> <p>The column searched can be changed to any column in <i>Ref. Table</i>.</p>
<i>Match Limit</i>	<p>The maximum number of results to display to the user. A default number (20) is specified, but this can be changed as required.</p>
<div style="border: 1px solid black; padding: 2px; display: inline-block;">  </div>	<p>Where <i>Type</i> is <code>attachment</code> it is possible to specify a Filter that restricts which records are returned by a search of <i>Ref. Table</i>.</p> <p>Click Add filter term to specify a <i>Column</i> and <i>Value</i> to restrict which records can be attached to.</p> <p>See <i>Attachment field</i> (page 67) for details.</p>

The screenshot shows the 'Sections' configuration interface. The main window is titled 'Sections' and contains a 'Section' configuration panel. The 'Section' panel has a 'Label' field set to 'Name' and an 'ID' field set to '1'. Below this is a 'Field' configuration panel. The 'Field' panel has an 'Identifier' field set to 'partytype' and a 'Column' field set to 'NamPartyType'. The 'Label' field is set to 'Party Type'. The 'Kind' dropdown is set to 'single' and the 'Type' dropdown is set to 'text'. The 'Lines' field is empty. The 'Access' dropdown is set to 'editable' and the 'Requirement' dropdown is set to 'mandatory'. The 'Lookup' field is set to 'Party Type' and the 'Level' field is set to '1'. The 'Default' field is empty. The 'Ref. Table' and 'Ref. Column' fields are empty. The 'Match Limit' and 'Image Width (Best Fit)' fields are empty. At the bottom of the 'Field' panel, there is a '+ Add filter term' button. At the bottom of the 'Section' panel, there is a '+ Add Field' button. At the bottom of the main 'Sections' window, there is a '+ Add section' button.


11. Click  to add another field to the section and configure as required.
12. Add as many fields to the Form as required.

If it makes sense to organise fields into logical groupings rather than adding them all to a single

section, click  to add another section to the Form.

For instance, in this example we could add a section called *Address* to group address details separately from the Party's name:

The screenshot displays the 'Sections' configuration interface. It features a hierarchical structure with sections and fields. The top section is 'Name (4 fields)'. Below it is a 'Section' with 'Label: Address' and 'ID: 2'. Underneath is a 'Field' with 'Identifier: street' and 'Column: AddPhysStreet'. The field configuration includes options for 'Kind: single', 'Type: text', 'Access: editable', 'Requirement: optional', 'Lookup', 'Level', 'Default', 'Ref. Table', 'Ref. Column', 'Match Limit', and 'Image Width (Best Fit)'. There is also a 'Filters' section with an 'Add filter term' button. At the bottom, there are buttons for '+ Add Field' and '+ Add section'.

 **Note:** Note that an ID has been assigned automatically for this second section. We learn how to reference Section IDs when we look at designing a Custom Layout for a Form (page 83).

In this screenshot the first section has been collapsed to conserve screen space: the number of fields in the collapsed section is indicated beside the section label (*Name* in this example).






Note: It is possible to drag and drop a field within a section (to change its position on the Form) and between sections. It is also possible to drag and drop sections in order to rearrange their order on the Form. While not necessary, it can be handy to collapse sections before dragging them.

13. Click  to save the Form.


If you did not specify an Identifier for the Form, it will be assigned automatically.

The Form undergoes validation. It will not be possible to save the Form until all mandatory fields have been completed.

14. When you're finished with the Form, save any changes and click:

	<p>Return to the Home screen.</p> <p>If you belong to more than one project, the Home screen lists all the projects you belong to. If you belong to one project only, the Home screen lists the elements of the project that you have access to.</p>
	<p>Return to the list of elements you have access to in the current project.</p> <div data-bbox="395 925 1449 1043" style="border: 1px solid black; background-color: #f0f0f0; padding: 5px;"> <p> Note: If you belong to one project only, the Home and Project buttons return you to the same location.</p> </div>

-OR-

- Click  to exit the application.

Kind: multiple

When a *Kind* of `multiple` is specified for a field:

Field
✕ ^

Identifier:	<input type="text" value="othernames"/>	Column:	<input type="text" value="NamOtherNames_1"/>
Label:	<input type="text" value="Other names"/>		
Kind:	<input style="border: none; border-bottom: 1px solid #ccc; width: 100%;" type="text" value="multiple"/>	Type:	<input style="border: none; border-bottom: 1px solid #ccc; width: 100%;" type="text" value="text"/>
Access:	<input style="border: none; border-bottom: 1px solid #ccc; width: 100%;" type="text" value="editable"/>	Requirement:	<input style="border: none; border-bottom: 1px solid #ccc; width: 100%;" type="text" value="optional"/>
Lookup:	<input type="text"/>	Level:	<input type="text"/>
Default:	<input type="text"/>		
Ref. Table:	<input type="text"/>	Ref. Column:	<input type="text"/>
Match Limit:	<input type="text"/>	Image Width (Best Fit):	<input type="text"/>

Filters

⊕ Add filter term

the user will be presented with a single data entry field in the first instance (*Other names* in this example):

🏠 📁
Gerard ↗

🔍
Add or edit Parties record
i


Name
^

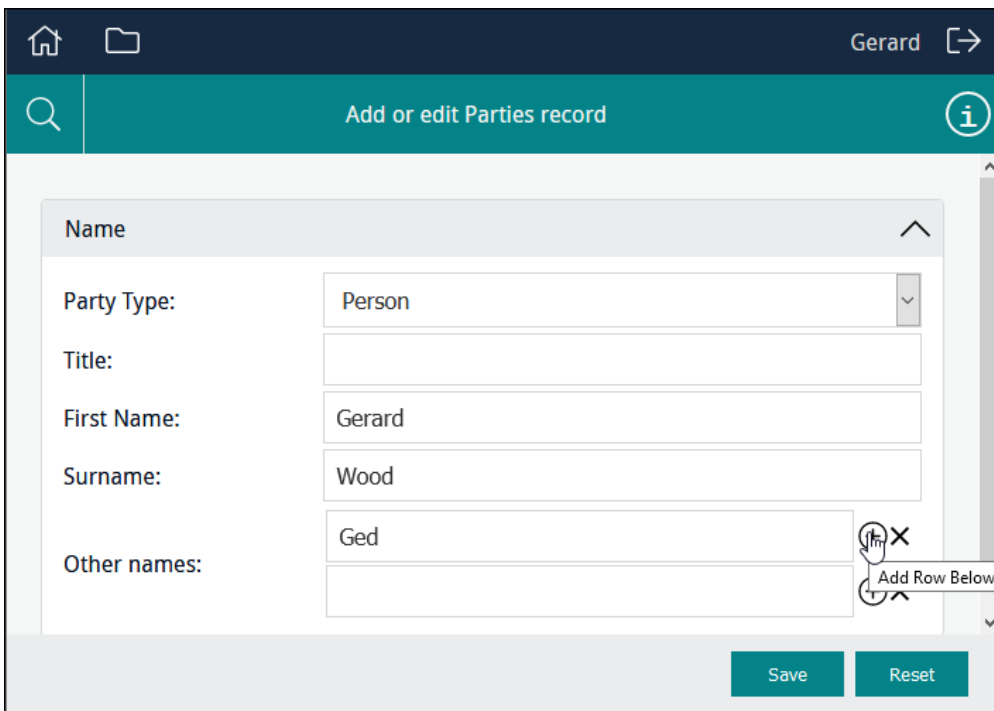
Party Type:	<input style="border: none; border-bottom: 1px solid #ccc; width: 100%;" type="text" value="select a value"/>
Title:	<input type="text"/>
First Name:	<input type="text"/>
Surname:	<input type="text"/>
Other names:	<input style="border: none; border-bottom: 1px solid #ccc; width: 100%;" type="text"/>


⊕ ✕

Save

Reset

Clicking the **Add Row**  icon beside a row will add a row immediately below it:



Selecting the **Remove Row**  icon beside a row will remove the row.



Note: When the back-end name of a table of values is specified in *Column*, *Kind* is automatically set to `multiple`. In the example above, `NamOtherNames_tab` is the back-end name of the *Other Names: (Person Details)* field.

Type: date

A date field is specified with a *Type* of date:

Field
✕ ^

Identifier:	<input type="text" value="birthdate"/>	Column:	<input type="text" value="BioBirthDate"/>
Label:	<input type="text"/>		
Kind:	<input type="text" value="single"/> ▾	Type:	<input type="text" value="date"/> ▾
Access:	<input type="text" value="editable"/> ▾	Requirement:	<input type="text" value="optional"/> ▾
Lookup:	<input type="text"/>	Level:	<input type="text"/>
Default:	<input type="text"/>		
Ref. Table:	<input type="text"/>	Ref. Column:	<input type="text"/>
Match Limit:	<input type="text"/>	Image Width (Best Fit):	<input type="text"/>

Filters
 (+) Add filter term

Date fields display a hint indicating the format of dates:

Birth Date:	(e.g. 31/12/2020)
-------------	-------------------

When the cursor enters the field, the hint disappears and a calendar pops-up:

Gerard

Add or edit Parties record

Name

Party Type:

Title:

First Name:

Surname:

Other names:

Birth Date:

April 2020

Su	Mo	Tu	We	Th	Fr	Sa
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30		

Save Reset

Type: selection

If *Column* has an associated Lookup List, selecting a *Type* of `selection` will present users with a read-only drop-list of values drawn from the Lookup List associated with *Column*. For example, a field such as *Party Type* in the Parties module, which has the back-end name *NamPartyType*, has an associated Lookup List called `Party Type`. In order to display values from the Lookup List associated with *Column*, it is not necessary to specify the name of the Lookup List in *Lookup*:

Field
✕ ^

Identifier:	<input type="text" value="partytype"/>	Column:	<input type="text" value="NamPartyType"/>
Label:	<input type="text" value="Party Type"/>		
Kind:	<input type="text" value="single"/> ▾	Type:	<input type="text" value="selection"/> ▾
Access:	<input type="text" value="editable"/> ▾	Requirement:	<input type="text" value="mandatory"/> ▾
Lookup:	<input type="text"/>	Level:	<input type="text"/>
Default:	<input type="text"/>		
Ref. Table:	<input type="text"/>	Ref. Column:	<input type="text"/>
Match Limit:	<input type="text"/>	Image Width (Best Fit):	<input type="text"/>

Filters
 (+) Add filter term

You would specify a different Lookup List in *Lookup* if you wanted users to be able to select a value from a different Lookup List in order to save that value to *Column*.

In this example the user is able to select from a read-only list of titles drawn from the *NamPartyType* column:

The screenshot shows a web application interface for adding or editing a Parties record. The interface is titled "Add or edit Parties record" and includes a search icon and a user profile icon labeled "Gerard". The form contains the following fields:

- Name
- Party Type: (dropdown menu showing "select a value", "Collaboration", "Group", "Organisation", "Person", "Position", "Ulan")
- Title:
- First Name:
- Surname:
- Other names:
- Birth Date:

At the bottom of the form, there are "Save" and "Reset" buttons.



Note: If a Lookup List is large, it might make sense to specify `Type: text` rather than `Type: selection` as `Type: text` filters the list of values presented to users as data is entered in a field, rather than displaying all available values in a drop-list.

Type: text

If *Column* has an associated Lookup List, selecting a *Type* of `text` will present users with a filtered read-only drop-list of values drawn from the Lookup List associated with *Column*. Filtering occurs as the user enters characters in the field. For example, a field such as *Title: (Person Details)* in the Parties module, which has the back-end name *NamTitle*, has a Lookup List called `Title`:

Field
✕ ^

Identifier:	<input type="text" value="title"/>	Column:	<input type="text" value="NamTitle"/>
Label:	<input type="text" value="Title"/>		
Kind:	<input type="text" value="single"/> ▾	Type:	<input type="text" value="text"/> ▾
Lines:	<input type="text"/>		
Access:	<input type="text" value="editable"/> ▾	Requirement:	<input type="text" value="optional"/> ▾
Lookup:	<input type="text"/>	Level:	<input type="text"/>
Default:	<input type="text"/>		
Ref. Table:	<input type="text"/>	Ref. Column:	<input type="text"/>
Match Limit:	<input type="text"/>	Image Width (Best Fit):	<input type="text"/>

Filters
 ⊕ Add filter term



Note: If the field specified in *Column* has an associated Lookup List, it is not necessary to specify a Lookup List in *Lookup*: the Lookup List associated with *Column* is automatically available to the Data Entry user.

In this example as the user enters characters in the field, a filtered read-only list of titles drawn from the *NamTitle* column is displayed:

The screenshot shows a web application interface for adding or editing a Parties record. The interface is titled "Add or edit Parties record" and includes a search icon, a home icon, and a user profile icon labeled "Gerard". The form contains the following fields:

- Name:** A header for the form.
- Party Type:** A dropdown menu with the text "select a value" and a warning icon.
- Title:** A text input field containing the letter "m".
- First Name:** A text input field containing the text "Miss".
- Surname:** A text input field containing the text "Mme.".
- Other names:** A text input field containing the text "Mr & Mrs".
- Birth Date:** A text input field containing the text "Mrs".

At the bottom of the form, there are two buttons: "Save" and "Reset".

You would specify a different Lookup List in *Lookup* if you wanted users to be able to select a value from a different Lookup List in order to save that value to *Column*.



Note: If a Lookup List is large, it might make sense to specify `Type: text` rather than `Type: selection` as `Type: text` filters the list of values presented to users as data is entered in a field, rather than displaying all available values in a drop-list.

Lookup

When a field is specified as having a *Type* of `selection` or `Text` and *Column* has an associated Lookup List, a user is presented with a list of read-only values drawn from *Column*.



Note: In order to display values from the Lookup List associated with *Column*, it is not necessary to specify the name of the Lookup List in *Lookup*.

The *Lookup* option is used:

- » In conjunction with `selection` or `text` and *Column*.

The back-end name of any Lookup List can be specified in *Lookup* in order to present its values to a user instead of values drawn from *Column*.

This could be useful, for instance, if *Column* includes a large number of entries and only a small number are required for the purposes of data entry.

For example, *AddPhysCountry* is associated with the `Location` Lookup List, but in this example the Form will include a drop-list of values drawn from the `Europe` Lookup List. A value selected by the user will be saved to the *AddPhysCountry* column:

Field
✕ ^

Identifier: <input type="text" value="country"/>	Column: <input type="text" value="AddPhysCountry"/>
Label: <input type="text" value="Country"/>	
Kind: <input type="text" value="single"/> ▼	Type: <input type="text" value="selection"/> ▼
Access: <input type="text" value="editable"/> ▼	Requirement: <input type="text" value="optional"/> ▼
Lookup: <input type="text" value="Europe"/>	Level: <input type="text"/>
Default: <input type="text"/>	
Ref. Table: <input type="text"/>	Ref. Column: <input type="text"/>
Match Limit: <input type="text"/>	Image Width (Best Fit): <input type="text"/>


Filters

⊕ Add filter term

- » Where no *Column* has been specified when using a Processing Script to perform some special processing (page 36).

Specify the back-end name of a Lookup List to present its values to a user. If the Lookup List is a multi-level hierarchy, use the *Level* field to specify which level of the hierarchy to present:

Field ✕ ^

Identifier: Column: 

Label:

Kind: ▼ Type: ▼

Access: ▼ Requirement: ▼

Lookup: Level:

Default:

Ref. Table: Ref. Column:

Match Limit: Image Width (Best Fit):

Filters

⊕ Add filter term

Attachment field

When an attachment field is specified in *Column*, a user is presented with a text box; when a value is entered in the text box, a search is performed in *Ref. Column* in the target module (*Ref. Table*) and matching results are listed (up to the number specified in *Match Limit*) from which the user can select a record. On saving the Form, an attachment is made from *Column* to the record selected in the target module (the IRN of the attachment record is saved in *Column*).

When an attachment field is specified in *Column*:

- » The table to which it attaches is automatically specified in *Ref. Table*, and *Ref. Column* is set to *SummaryData*; a default number (20) is added to *Match Limit*.
- » It is possible to specify a Filter to restrict which records are returned by the search of the target module.

In this simple example, users will be presented with a field that attaches to the Parties module:

Field
✕ ^

Identifier: <input type="text" value="association"/>	Column: <input type="text" value="AssAssociationRef_"/>
Label: <input type="text" value="Association"/>	
Kind: <input type="text" value="multiple"/> ▾	Type: <input type="text" value="attachment"/> ▾
Access: <input type="text" value="editable"/> ▾	Requirement: <input type="text" value="optional"/> ▾
Lookup: <input type="text"/>	Level: <input type="text"/>
Default: <input type="text"/>	
Ref. Table: <input type="text" value="eparties"/>	Ref. Column: <input type="text" value="SummaryData"/>
Match Limit: <input type="text" value="20"/>	Image Width (Best Fit): <input type="text"/>

Filters

⊕ Add filter term ☞

As letters are entered into the field, records are progressively filtered and all matching records in the Parties module will be listed:

The screenshot shows a web application interface for managing parties. The main header is 'Add or edit Parties record'. The form contains the following fields:

- Name** (expandable section):
 - Party Type: Person
 - Title: Dr
 - First Name: Gerard
 - Surname: Wood
 - Other names: Ged, Iggy (with add/remove icons)
 - Birth Date: (empty)
 - Country: select a value (dropdown)
 - Association: pea (with dropdown menu open showing: Pearce, Leslie; Pearlman, Betty; Pearson, George; **Peasley, Colin**; Yale Peabody Museum of Natural History)

At the bottom right of the form are 'Save' and 'Reset' buttons. The 'Save' button is highlighted with a red box in the original image.

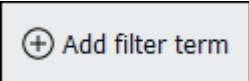
When the user selects an entry in the list and clicks Save, the attachment is made to a person or organisation associated with the current Parties record (the attachment record's IRN is saved to *Column*).

When an attachment field is specified in *Column*:

- » *Type* is automatically set to `attachment`.
- » *Ref. Table* is automatically set to the back-end name of the table to which the attachment field specified in *Column* attaches.
- » *Ref. Column* is automatically set to `SummaryData`. This is the column searched for a match when a user enters values in the Form. The column searched can be changed to any column in *Ref. Table*.
- » *Match Limit* is automatically set to the default number of results (20) to display to the user. This number can be changed as required.

Rather than presenting users with all possible records in the Parties module, it is possible to add a filter that limits which records are returned when the user enters characters into the field.



To add a filter, click  and specify a *Column* and *Value* to restrict which records are returned when the user enters characters into the field. In this example, only records where *Party Type* = *Organisation* will be listed:

Field
✕ ^

Identifier:

Column:

Label:

Kind: ▾

Type: ▾

Access: ▾

Requirement: ▾

Lookup:

Level:

Default:

Ref. Table:

Ref. Column:

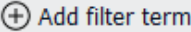
Match Limit:

Image Width (Best Fit):

Filters

Column:

Value: ✕



EMu (version 5.0 onwards): escaped double quotes are required around any value that includes a space, *New Zealand* for example, in order to match a value exactly. For example, to add a filter that will only list records for Parties in *New Zealand*, you would specify:

- » *Column* = `AddPhysCountry`
- » *Value* = `\ "New Zealand\"`

Multimedia fields

It is possible to display an image on a Form, and to enable users to upload multimedia accessible to their device, and to capture an image with their device and upload it.

Four image *Kinds* are available:

» `image` (page 71)

Display an image in a Form.

A typical use might be to display a scan or other image which contains information that a user would locate and key in to other fields in the Form.

» `capture` (page 75)

Enable users to locate multimedia of any kind that is accessible to their device and upload it.



Note: `capture` does not allow users to take a photo with their device and upload it.

» `image-capture` (page 79)

Enable users to locate images accessible to their computer / device and upload it via the Form.



Note: `image-capture` may enable users to take a photo with their device and upload it, depending on their mobile device.

» `mobile-image-capture` (page 81)

Enable a user with a mobile device to capture an image with the device's camera and upload it.

Keep in mind

You typically build forms that allow users to add or edit records in modules such as the Catalogue or Parties. On these forms you can include Multimedia fields and users will be able to add or remove attachments to Multimedia records. Note however that this does not allow users to delete records from the Multimedia module. When users upload multimedia when editing Catalogue records, for example, this will create a record in the Multimedia module for the uploaded resource and attach that record to the record they are editing. A Form would need to be designed specifically for the purpose of editing records in the Multimedia module in order to alter Multimedia records themselves.

Kind: image

When specifying a *Kind* of `image` our objective is to display an image on the Form, perhaps to allow a user to locate information recorded on the image and capture it in fields on the Form.

When a *Kind* of `image` has been selected, we point *Column* at a field that holds details about the image:

Home Folder Gerard ↔

Title:

Description:

Identifier: Table:

Processing script: Deferrable:

✓ Permissions ^

Everyone:

[+ Add permission](#)

📁 Sections ^

Section ✕ ^

Label: ID:

Field ✕ ^

Identifier: Column:

Label:

Kind: Type:

Access: Requirement:

Lookup: Level:

Default:

Ref. Table: Ref. Column:

Match Limit: Image Width (Best Fit):

Filters

[+ Add filter term](#)



Note: When adding an image control (page 70) to a Form, it may be better not to include a Field label in order to maximise the size of the image control. This is particularly important when using a custom layout (page 83) with columns.

In this example, *Column* points to *MulMultiMediaRef_tab*, a text field in the Catalogue module which holds details about images used during data capture, including:

- » The IRN of the Multimedia record for the image.
- » Its zoom state (when the image is re-displayed, it will be zoomed accordingly).
- » Any polygons marked on the image.

The field which holds details about images used during data capture is populated with these details when an image is uploaded using Sapphire. See *Kind: image-capture* (page 79) for details.



Note: The field can also be populated with details about images using various back-end scripts.

Having added an image to the Form, we could then add fields necessary to record any details available on the image:

Gerard

Label Digitisation

Image
^

+

-

Details
^

Accession Lot:

Summary:

Description
^

Description:

Save

Reset

Kind: capture

When specifying a *Kind* of `capture` our objective is to enable users to locate any type of multimedia file accessible to their computer / device and upload it.



Note: `capture` does not enable users to take a photo with their device and upload it via the Form.

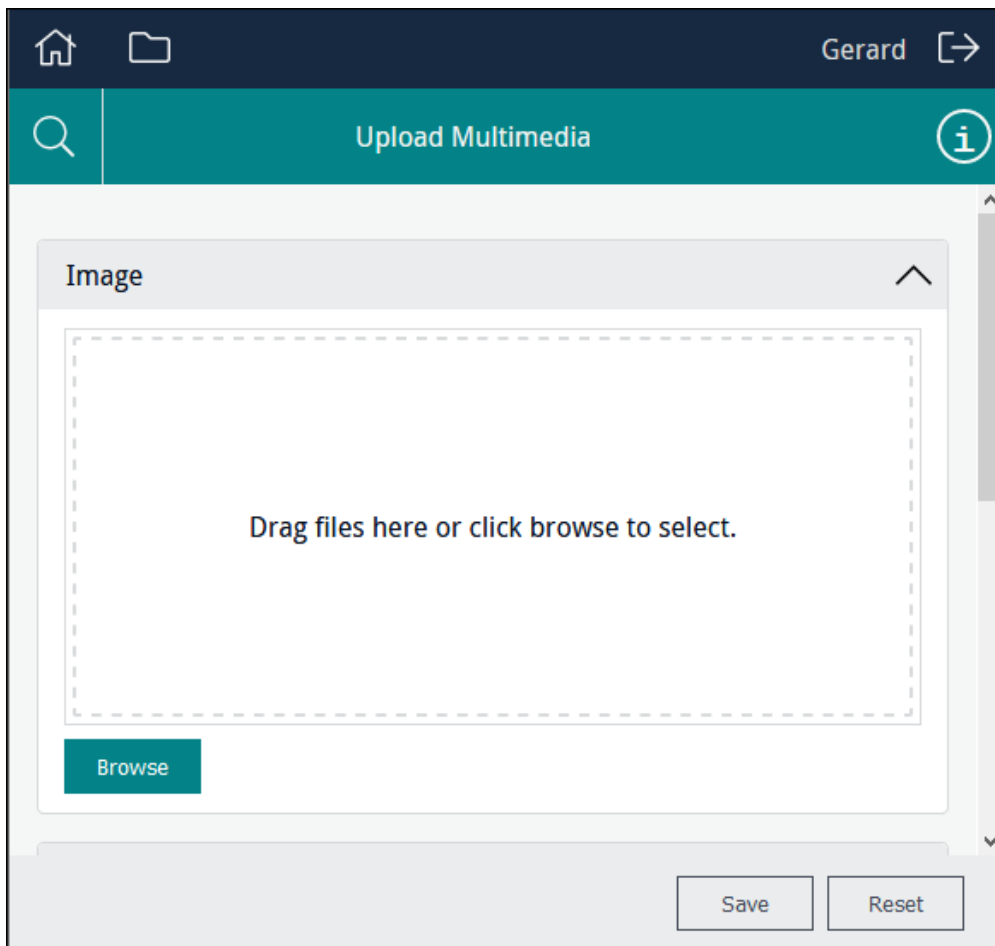
When a *Kind* of `capture` has been selected, we point *Column* at a field that will hold details about the image when the image is uploaded using the Form:

Field
✕ ^

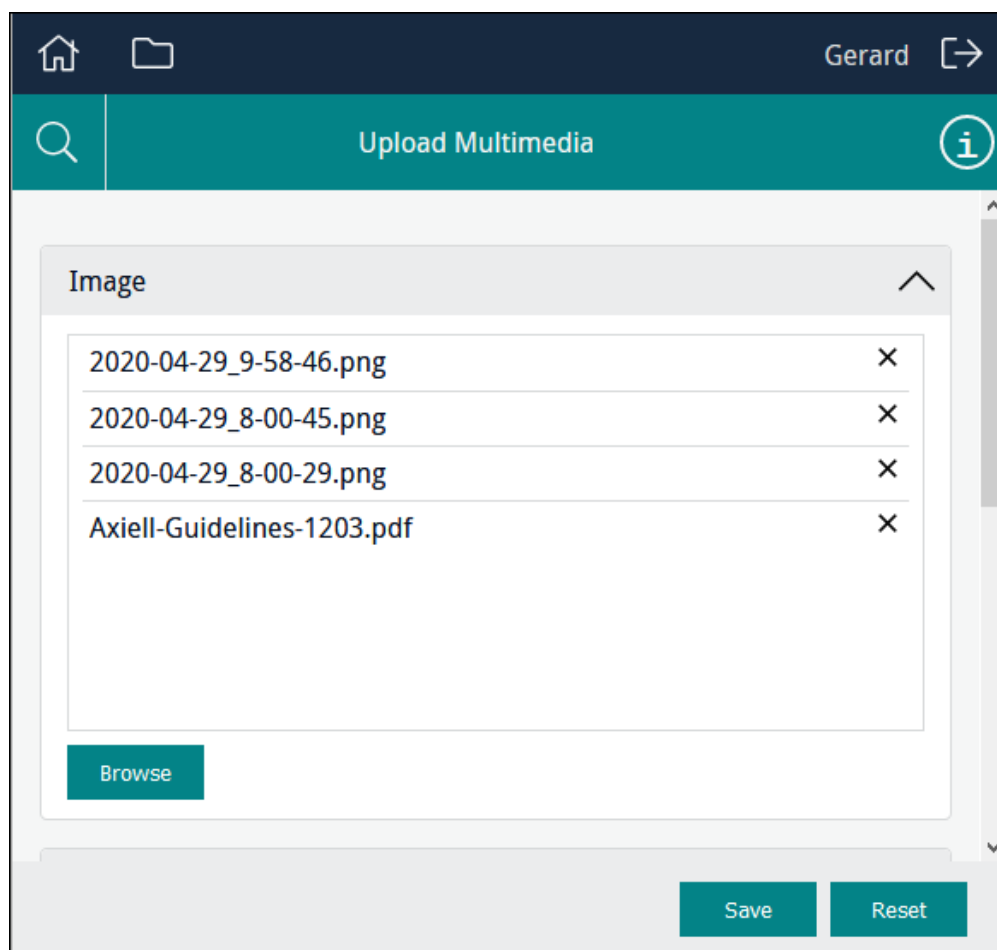
Identifier:	<input type="text" value="img"/>	Column:	<input type="text" value="MulMultiMediaRef_"/>
Label:	<input type="text"/>		
Kind:	<input type="text" value="capture"/> ▼	Type:	<input type="text" value="text"/> ▼
Access:	<input type="text" value="editable"/> ▼	Requirement:	<input type="text" value="optional"/> ▼
Lookup:	<input type="text"/>	Level:	<input type="text"/>
Default:	<input type="text"/>		
Ref. Table:	<input type="text"/>	Ref. Column:	<input type="text"/>
Match Limit:	<input type="text"/>	Image Width (Best Fit):	<input type="text"/>


Filters
⊕ Add filter term

In this example, *Column* points to `MulMultiMediaRef_tab`, a text field in the Catalogue module which holds details about multimedia uploaded during data entry:



Users can drag one or more files onto the control or click **Browse** to navigate to a file and add it to the control:



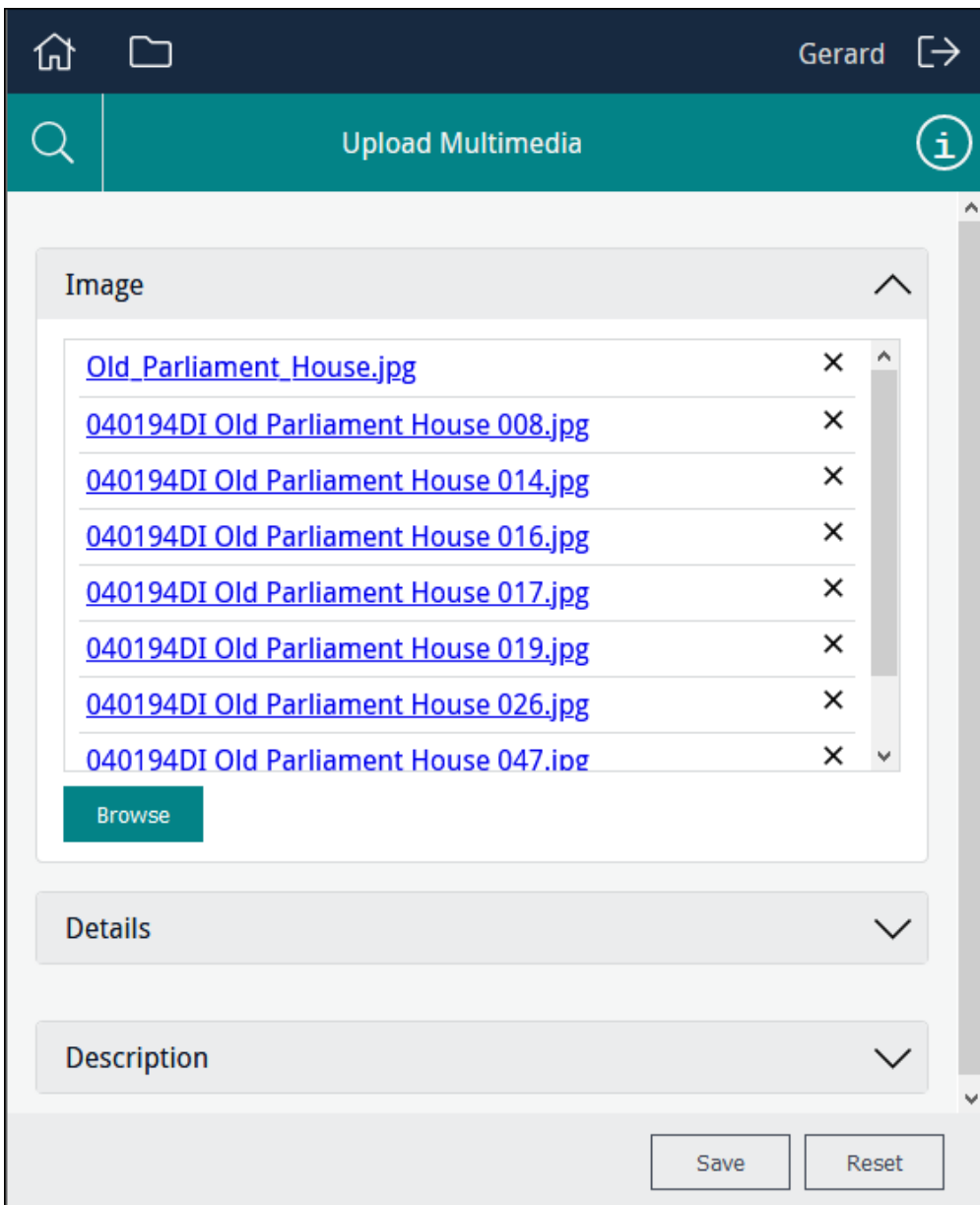
A file can be removed from the list by clicking  beside the file name.



Note: The `capture` control only lists the name of multimedia files, it does not preview them.

When the Form is saved, a record will be created in the Multimedia module for each of the uploaded files. When using a Form to create or edit a record, in the Catalogue module for instance, the newly created Multimedia records will be attached to the current record.

If the Form allows users to search a module for existing records, any multimedia already attached to the record will be listed:



Note: A multimedia resource listed in the `capture` control can be downloaded. How this is done will depend on the browser being used, and how the browser has been configured to deal with links. For instance, it may be possible to download the resource by left-clicking the link, or it may be necessary to right-click a link and select **Save link as**.

Clicking  beside a file name will remove the attached Multimedia record from the current record.

Kind: image-capture

When specifying a *Kind* of `image-capture` our objective is to enable users to locate an image file accessible to their computer / device and upload it.

When the record is saved, the image is uploaded to the EMu Multimedia repository and attached to the record being created (a Catalogue record in the example below).

When a *Kind* of `image-capture` has been selected, we point *Column* at a field that will hold details about the image when the image is uploaded using the Form:

Field
✕ ^

Identifier:	<input type="text" value="img"/>	Column:	<input type="text" value="MulMultiMediaRef_"/>
Label:	<input type="text"/>		
Kind:	<input type="text" value="image-capture"/> ▾	Type:	<input type="text" value="text"/> ▾
Access:	<input type="text" value="editable"/> ▾	Requirement:	<input type="text" value="optional"/> ▾
Lookup:	<input type="text"/>	Level:	<input type="text"/>
Default:	<input type="text"/>		
Ref. Table:	<input type="text"/>	Ref. Column:	<input type="text"/>
Match Limit:	<input type="text"/>	Image Width (Best Fit):	<input type="text"/>

Filters

⊕ Add filter term



Note: When adding an image control (page 70) to a Form, it may be better not to include a Field label in order to maximise the size of the image control. This is particularly important when using a custom layout (page 83) with columns.

In this example, *Column* points to `MulMultiMediaRef_tab`, a text field in the Catalogue module which holds details about images captured during data entry, including:

- » The IRN of the Multimedia record for the image.
- » Its zoom state (when the image is re-displayed, it will be zoomed accordingly).
- » Any polygons marked on the image.



Note: This field can also be populated with details about images using various back-end scripts.

As with an `image-capture` field, we might also include fields to capture any other details required at this stage of the workflow:

The screenshot shows a mobile application interface for 'Label Digitisation'. The top navigation bar is dark teal and contains a home icon, a folder icon, the name 'Gerard', and a share icon. Below this is a teal bar with a search icon, the title 'Label Digitisation', and an information icon. The main content area is divided into two sections: 'Image' and 'Details'. The 'Image' section has a 'Select image' button, a large empty image area with a plus and minus button on the left and a toolbar with icons for drawing, erasing, zooming, panning, and deleting on the right. The 'Details' section contains two text input fields labeled 'Accession Lot:' and 'Summary:'. At the bottom of the screen are 'Save' and 'Reset' buttons.



Note: `image-capture` may enable users to take a photo with their device and upload it, depending on their mobile device. `mobile-image-capture` is designed to work with mobile devices.

Kind: mobile-image-capture

When specifying a *Kind* of `mobile-image-capture`, users with a tablet / mobile device are able to capture an image with the device's camera and upload it using the Form.

When the record is saved, the image is uploaded to the EMu Multimedia repository and attached to the record being created (a Catalogue record for example).

When a *Kind* of `mobile-image-capture` has been selected, we point *Column* at a field that will hold details about the image when the image is uploaded using the Form:

The screenshot shows a 'Field' configuration window with the following fields:

- Identifier:
- Column:
- Label:
- Kind: (dropdown)
- Type: (dropdown)
- Access: (dropdown)
- Requirement: (dropdown)
- Lookup:
- Level:
- Default:
- Ref. Table:
- Ref. Column:
- Match Limit:
- Image Width (Best Fit):

At the bottom, there is a 'Filters' section with a button labeled '+ Add filter term'.

Note: When adding an image control (page 70) to a Form, it may be better not to include a Field label in order to maximise the size of the image control. This is particularly important when using a custom layout (page 83) with columns.

In this example, *Column* points to `MulMultiMediaRef_tab`, a text field in the Catalogue module which holds details about images captured during data entry, including:

- » The IRN of the Multimedia record for the image.
- » Its zoom state (when the image is re-displayed, it will be zoomed accordingly).
- » Any polygons marked on the image.

Note: This field can also be populated with details about images using various back-end scripts.

As for the other image kinds, we might include fields on the Form to capture any other details required at this stage of the workflow.







The Deferrable system

The *Deferrable* option is intended for wireless devices and affects the order in which records are processed in the event that wireless connectivity is lost. By default (*Deferrable* = `No`): if wireless connectivity is lost, record save will fail and an error message will be generated.

When *Deferrable* is set to `Yes` and a device is unable to save a record due to loss of wireless connectivity:

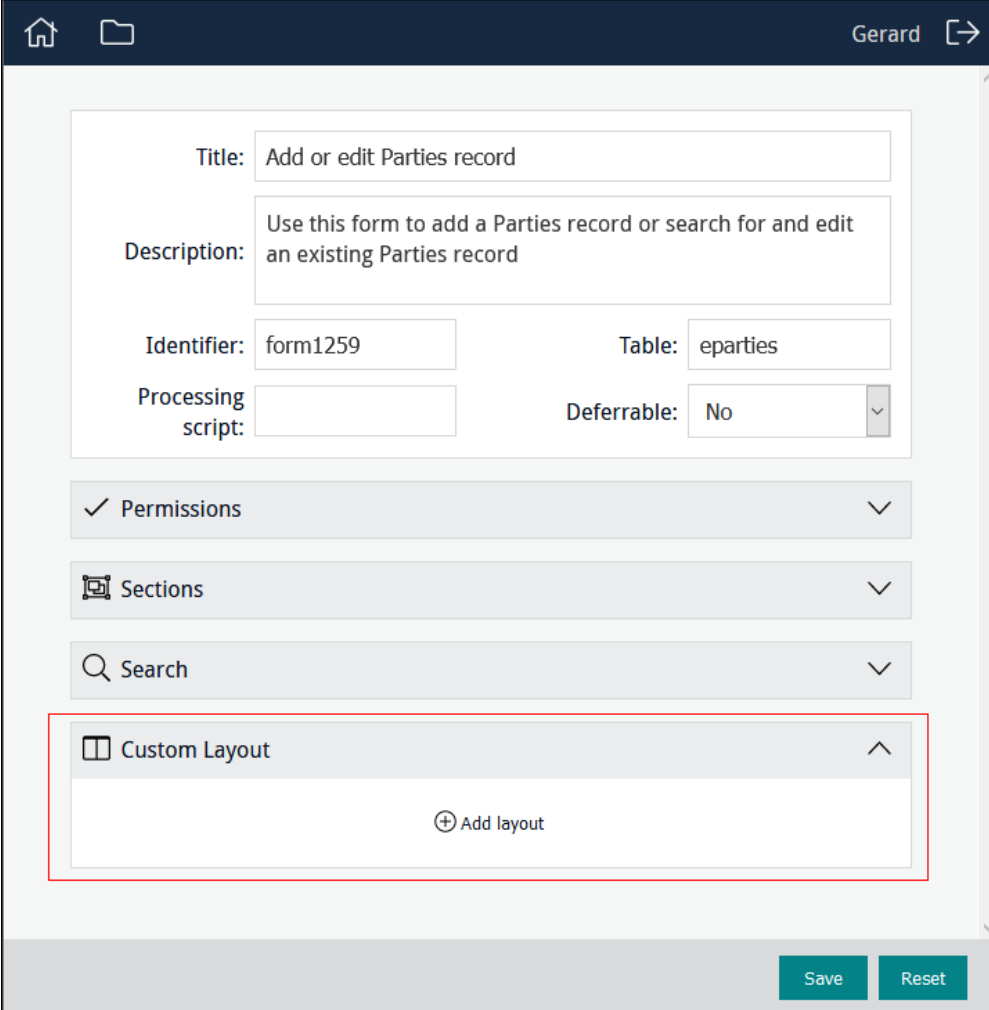
- » A queue of unsaved records is created.
- » Each unsaved record is added to the end of the queue. This ensures that when a connection is re-established and the records are saved, they are saved in the order in which they were created (oldest first).
- » The system will attempt to save the queued records at regular intervals until all records are removed from the queue:
 - » If a save attempt is successful, the record is removed from the front of the queue and an attempt is made to save the next record, and so on until the queue is empty.
 - » If a save attempt fails for reasons other than connectivity, an error message is recorded and the save data is removed from the front of the queue, and an attempt is made to save the next record, and so on until the queue is empty.
- » Three icons are added to Sapphire:



	<p>WiFi connection status. Icon is filled when there is a connection. When connection is lost, the icon displays as:</p> 
	<p>Error notification. When saving records fails for any reason other than connectivity, the icon display as:</p>  <p>Click the icon to view error messages.</p>
	<p>When there are records waiting to be saved, the top bulb of timer icon is filled:</p>  <p>At regular intervals the system will attempt to save any queued records. When records are successfully saved, the timer reverts to its original form (empty top bulb).</p>

How to customise the layout of a Form

The Custom Layout section of the Form Designer enables designers to customise the layout of a Form by assigning sections to columns:



The screenshot shows the Form Designer interface for a form titled "Add or edit Parties record". The form is configured with the following details:

- Title:** Add or edit Parties record
- Description:** Use this form to add a Parties record or search for and edit an existing Parties record
- Identifier:** form1259
- Table:** eparties
- Processing script:** (empty)
- Deferrable:** No

The interface includes several sections for configuration, each with a checkmark and a dropdown arrow:

- ✓ Permissions
- Sections
- Search
- Custom Layout** (highlighted with a red box)

The Custom Layout section is currently empty and contains a button labeled "Add layout". At the bottom right of the interface, there are "Save" and "Reset" buttons.

By default sections are presented one above the other in the order in which they are arranged in the Form Designer. This sort of layout is not always the most useful however. If a Form includes an image control, it may be more useful to present the image and data entry fields side-by-side in two columns. If there are lots of data entry fields, it would be ideal for the columns to be independently scrollable so that the image is always visible:

Gerard

Add or edit Parties record

Name

Party Type: !

Title:

First Name:

Surname:

Other names: +X

Birth Date:

Country: v

Association: +X

Image Capture

Select image

+
-

Two column layout

Biography

Column 1 has a scrollbar

Save Reset

With the Custom Layout section of the Form Designer it is possible to:

- » Specify one or more columns and assign Sections to columns.
- » Specify which column(s) are scrollable.

Customising the layout of a data entry Form



Note: It is only possible to design a custom layout with the Form Designer (i.e. for data entry Forms). The Editors and Statistics Designers do not enable custom layouts.

For this example we want two columns, with an image in the right column and data entry fields in the left. Sections are assigned to columns, so the first step in designing a custom layout is to add sections and fields to the Form (page 44).

In this example we create a layout with two columns:

1. The first column will hold all the data entry fields.
2. The second column will have a single image field

When we add a section to a Form, an ID is automatically assigned to the section:

The screenshot displays the Form Designer interface with three sections, each with a label and an ID field. The ID fields are highlighted with red boxes:

- Section 1:** Label: Name, ID: 1
- Section 2:** Label: Image, ID: 2
- Section 3:** Label: Biography, ID: 3

Each section also includes a field for adding more fields (e.g., Party Type, Title, Photo, biography) and an 'Add Field' button. At the bottom, there is an 'Add section' button.

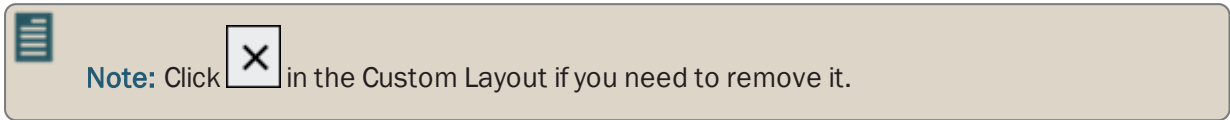
We assign these section IDs to columns to determine where sections will display on a Form.

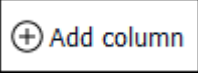
Sapphire Guide

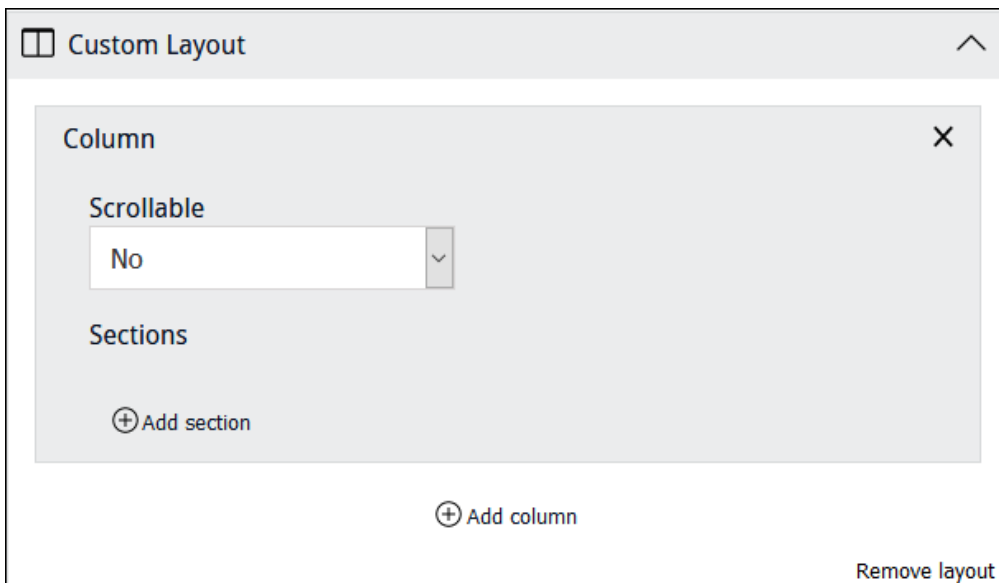
When you have added the required sections and fields to your Form:

1. Click  in the Custom Layout section of the Form Designer.

A Custom Layout is added to the Form.



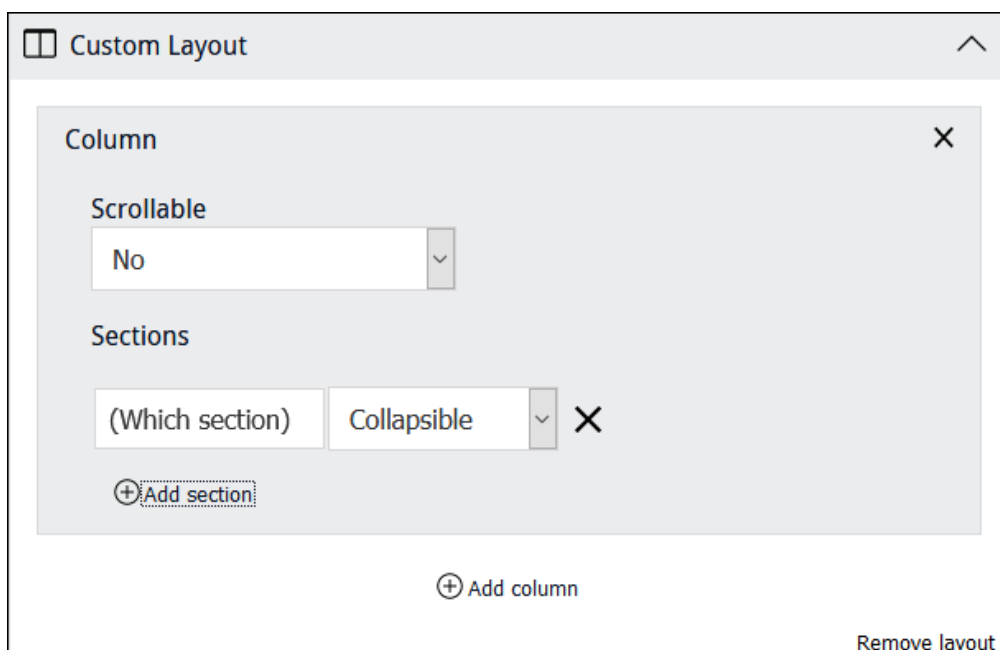
2. Click  to add a column to the Form:

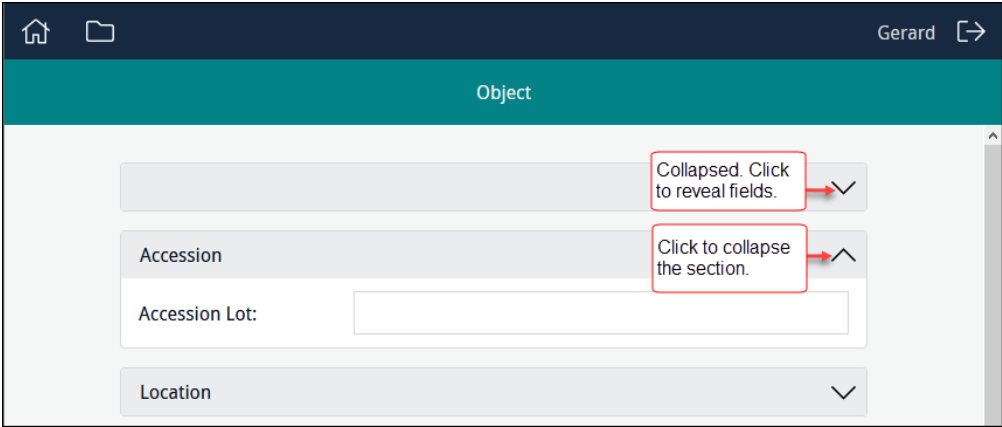


Next we assign sections to a column.

3. Click .

For each section added to a column, two pieces of information are required:



(Which section)	<p>Enter the ID assigned automatically when the Section was created.</p> <div style="border: 1px solid #ccc; padding: 5px; background-color: #f9f9f9;"> <p>Note: A section can only display once on a Form.</p> </div>
Collapsible / Not collapsible	<p>Specifies whether a section can be toggled open / closed (collapsed or not). By default, sections are collapsible and the Expand / Collapse toggle will display in the section's Title bar:</p> 

4. Assign as many sections to a column as required. Adjust the order of sections within a column by dragging and dropping a section.
5. Add one or more columns.

The order of columns in the Form Designer determines their display order on the Form: the first (top) column in the Form Designer will display on the far left of the Form, the last (bottom) on the far right. Adjust their order by dragging and dropping them.



The custom layout shown here (page 83) is configured as:

The screenshot shows a 'Custom Layout' window with two columns. The top column is titled 'Column' and has a close button (X) in the top right. It contains a 'Scrollable' dropdown menu set to 'Yes' and a 'Sections' list with two items: '1' and '3', both set to 'Collapsible' with their own close buttons (X). Below the sections is an 'Add section' button. The bottom column is also titled 'Column' with a close button (X) in the top right. It has a 'Scrollable' dropdown menu set to 'No' and a 'Sections' list with one item: '2', set to 'Collapsible' with a close button (X). Below the sections is an 'Add section' button. At the bottom center of the window is an 'Add column' button, and at the bottom right is a 'Remove layout' button.

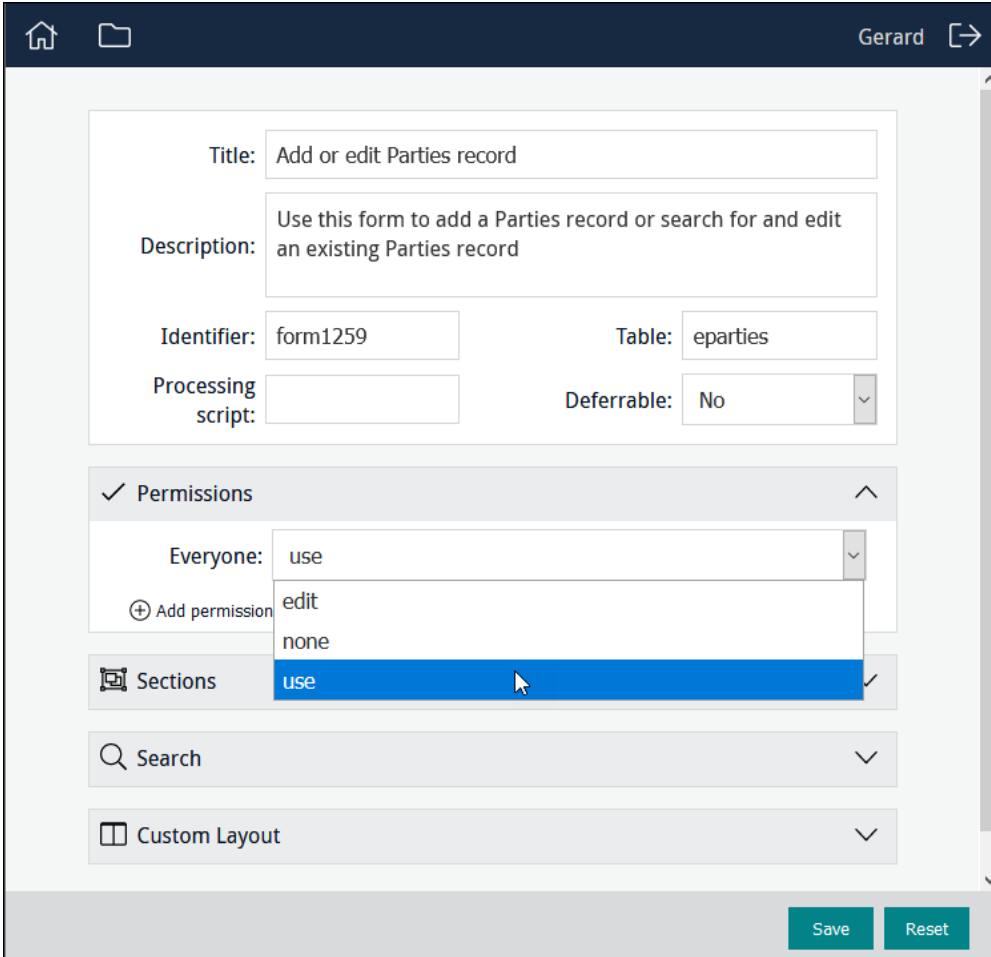
6. Click  to save the Form.

How to specify permissions for project elements

Permissions for project elements (Forms, Editors, Statistics) are:

none	When a project element is created this is the default state for Everyone: the element is not available to users and will not be listed when they log in.
use	When assigned, the project element will be listed when an authorised user logs in. The user can use the project element but not modify it.
edit	When assigned, the project element will be listed when an authorised user logs in. The user can use the project element, modify and delete it. When the project element is listed in the Dashboard, the edit  and delete  icons will display beside its name.

To change everyone's access permissions to a project element, select the appropriate permission from the *Everyone* drop-list:



The screenshot shows the configuration page for a project element titled "Add or edit Parties record". The description is "Use this form to add a Parties record or search for and edit an existing Parties record". The identifier is "form1259" and the table is "eparties". The processing script is empty, and the deferrable status is "No".

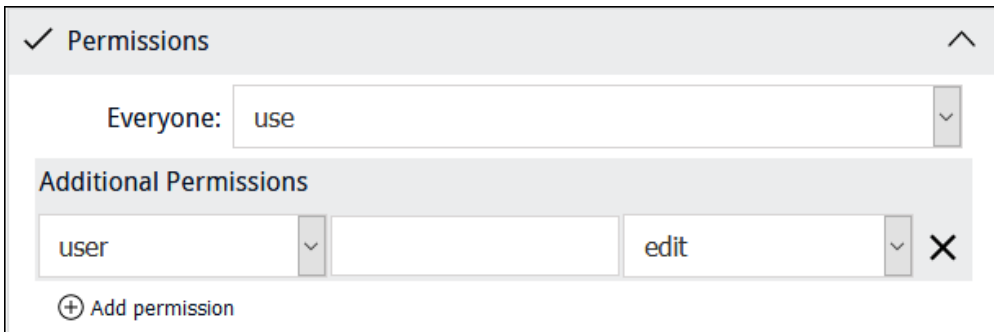
The **Permissions** section is expanded, showing a dropdown menu for "Everyone" with the following options: "use", "edit", and "none". The "use" option is selected and highlighted in blue. Below the dropdown, there is a "+ Add permission" button.

Other sections visible include "Sections" (with "use" selected), "Search", and "Custom Layout". At the bottom right, there are "Save" and "Reset" buttons.

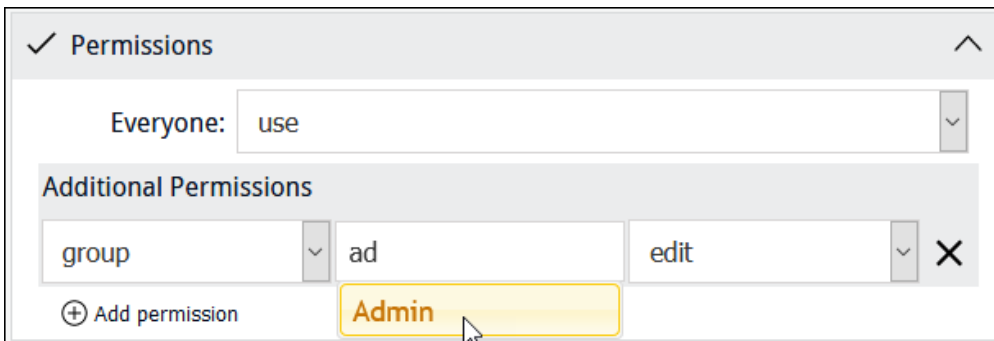
To change access permissions to a project element for a specific user or group:

1. Click 

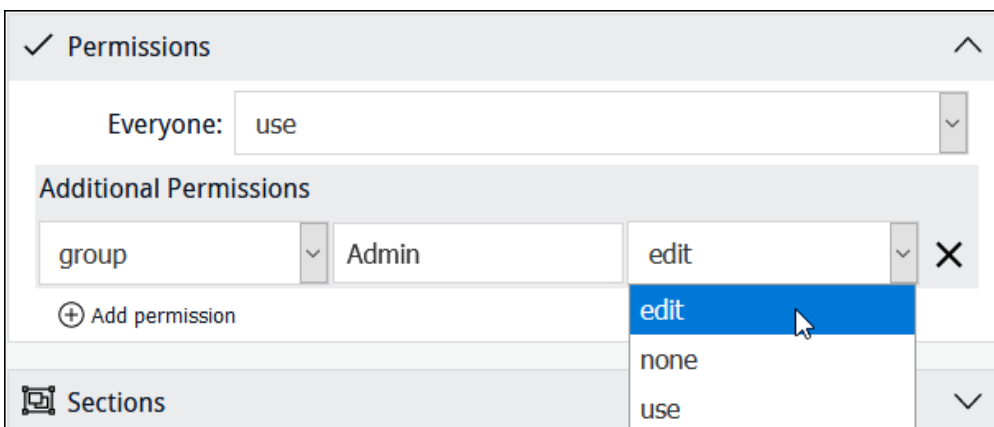
Additional *Permissions* fields display:




2. Select `user` or `group` from the first drop-list.
3. Begin keying a user or group name in the second field and select a name from the list of suggestions that will display:



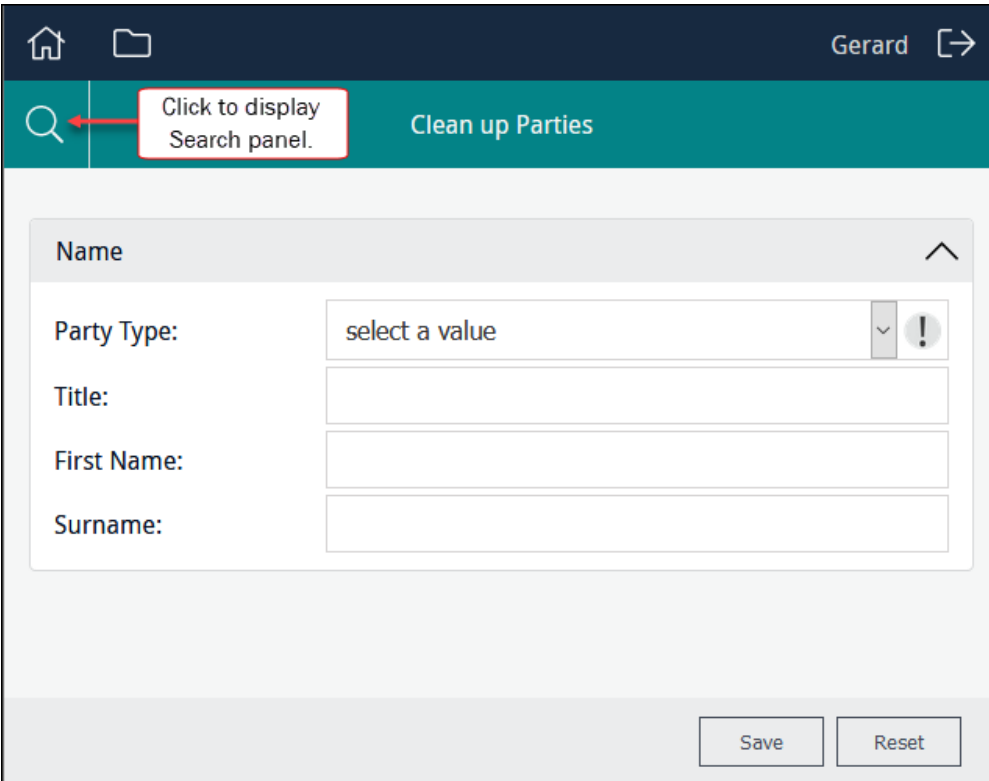
4. Select a permission from the *Permissions* drop-list:



Note: Click  beside an assigned user or group permission to remove it.

How to add a Search box in a Form

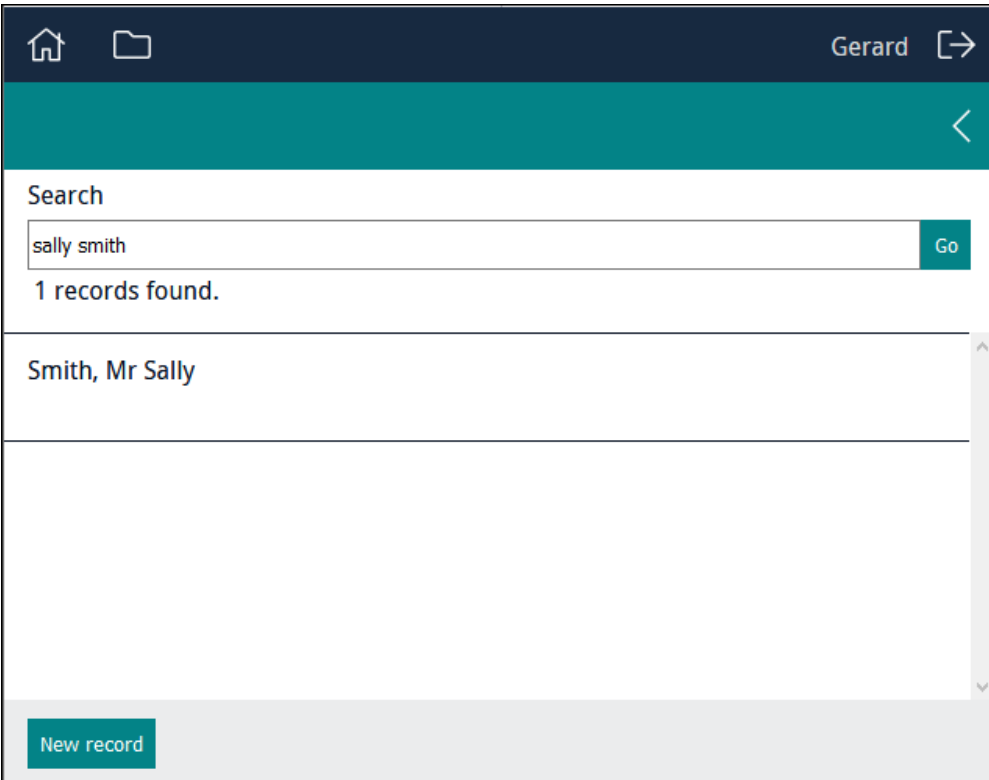
It is possible to include a Search section in a Form, enabling users to search for a record and update it:



The screenshot shows a mobile application interface. At the top, there is a dark blue header with a home icon, a folder icon, and the name 'Gerard' with a right-pointing arrow. Below the header is a teal bar with a search icon on the left and the text 'Clean up Parties' on the right. A red box with a white background and a red arrow points to the search icon, containing the text 'Click to display Search panel.' Below the teal bar is a form titled 'Name' with a chevron icon in the top right corner. The form contains four input fields: 'Party Type:' with a dropdown menu showing 'select a value' and a warning icon; 'Title:'; 'First Name:'; and 'Surname:'. At the bottom of the form are two buttons: 'Save' and 'Reset'.




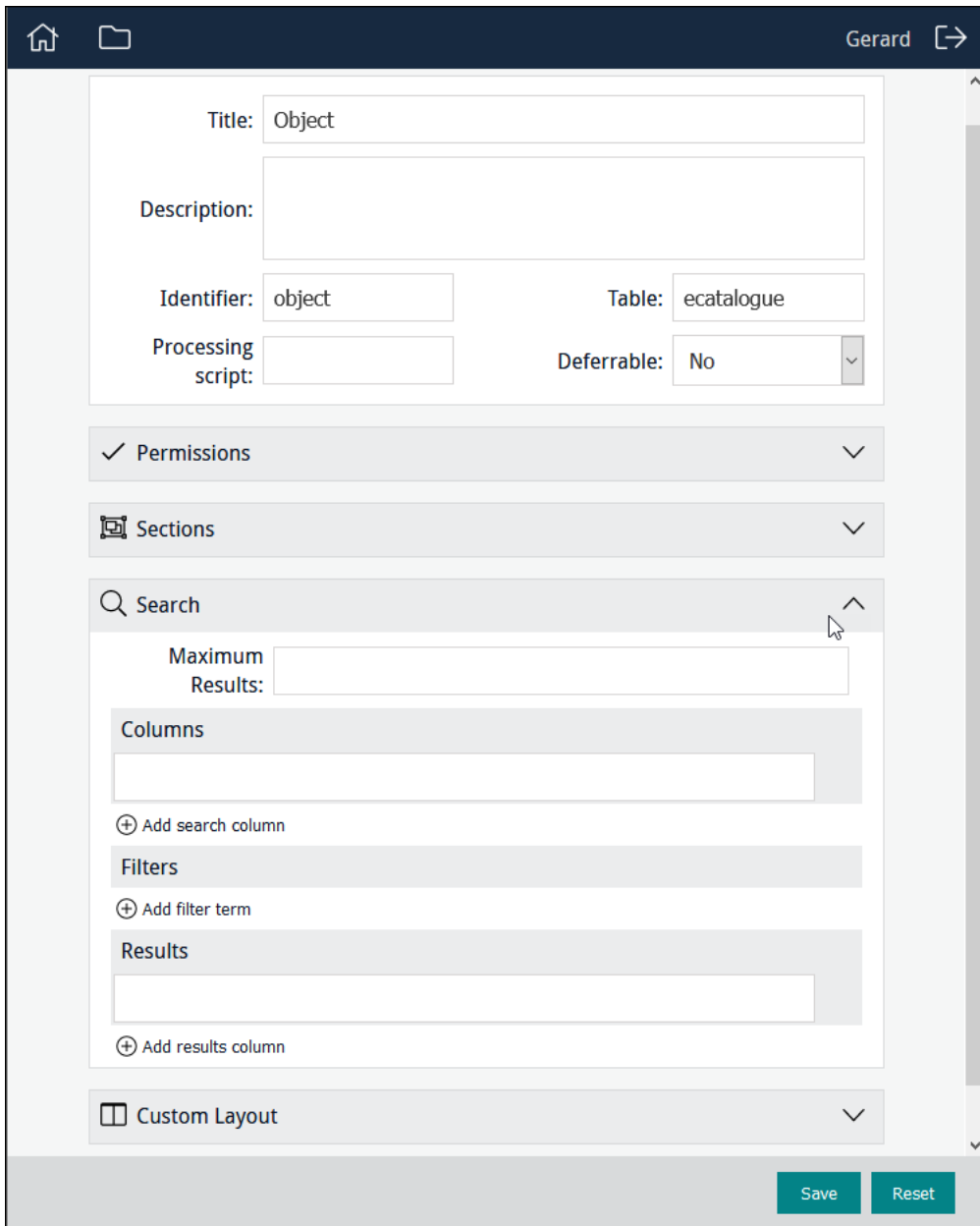
A Search panel will slide out from the left when the **Search** icon is clicked:




The screenshot shows the same mobile application interface, but with the search panel expanded. The teal bar now has a left-pointing chevron icon. The search panel is a white box with a teal header containing the word 'Search'. Below the header is a search input field containing the text 'sally smith' and a teal 'Go' button. Below the input field, it says '1 records found.' Below this is a list of search results, with the first result being 'Smith, Mr Sally'. At the bottom of the search panel is a teal button labeled 'New record'.

To add a Search section to a Form:

1. Add data entry fields to your Form.
2. Expand the Search section by clicking  in the Search banner:

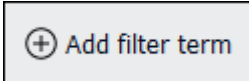


The screenshot shows the Sapphire form editor interface. At the top, there is a navigation bar with a home icon, a folder icon, and the name 'Gerard' with a refresh icon. Below this, the form editor displays several sections: 'Title' (Object), 'Description' (empty text area), 'Identifier' (object), 'Table' (ecatalogue), 'Processing script' (empty text area), and 'Deferrable' (No). Below these are expandable sections: 'Permissions', 'Sections', 'Search', and 'Custom Layout'. The 'Search' section is expanded, showing a 'Maximum Results' field, a 'Columns' field, and 'Filters' and 'Results' sections. The 'Columns' field is currently empty. Below the 'Columns' field is a button labeled '+ Add search column'. Below the 'Filters' section is a button labeled '+ Add filter term'. Below the 'Results' section is a button labeled '+ Add results column'. At the bottom right of the form editor, there are 'Save' and 'Reset' buttons.

3. *Maximum Results* is the maximum number of records that will be returned and displayed to users. Leave blank to return all records that match a user's search or enter a number to limit the number of records displayed.
4. Click  to specify which column (or columns) will be searched by the user.
5. In the *Columns* field, start keying a field name and select the required field when it displays:

The screenshot shows a search interface with the following sections:

- Search:** A magnifying glass icon and the word "Search".
- Maximum Results:** A text input field containing "10".
- Columns:** A text input field containing "summ". Below it, a yellow highlight is over the "SummaryData" option.
- Filters:** A section with a "+ Add filter term" button.
- Results:** A large empty text area.
- Bottom:** A "+ Add results column" button.

6. If required, add a filter by clicking .

A filter allows you to refine which records are returned by a user's search.

As this Form is intended for updating the Parties record of people (rather than organisations), we specify that only records where *Party Type* = `Person` will be returned by the search:

The screenshot shows the search interface with a filter applied:

- Search:** A magnifying glass icon and the word "Search".
- Maximum Results:** A text input field containing "10".
- Columns:** A text input field containing "SummaryData". Below it, a "+ Add search column" button.
- Filters:** A section showing a filter: "Column: NamPartyTy" and "Value: Person" with a close button (X).
- Results:** A large empty text area.
- Bottom:** A "+ Add results column" button.



Note: EMu (version 5.0 onwards): escaped double quotes are required around any search value that includes a space, New Zealand for example, in order to match a value exactly: e.g. `\ "New Zealand\ "`

7. Click + Add results column to specify which column(s) of data will display when search results are returned.
8. In the *Results* field, start keying a field name and select the required field when it displays:

The screenshot shows a search interface with the following sections:

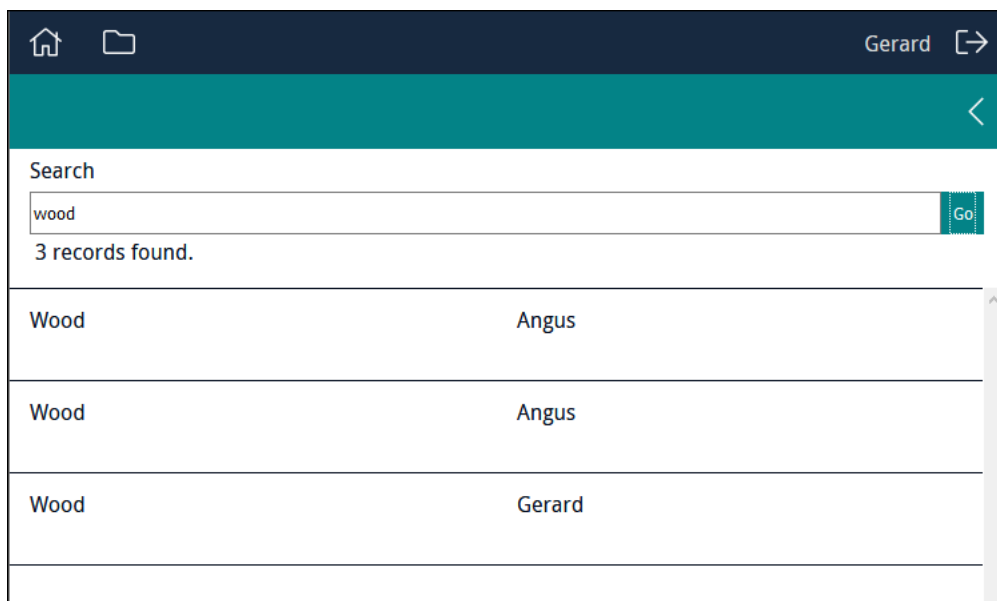
- Search:** A search bar with a magnifying glass icon and an upward arrow.
- Maximum Results:** A text input field containing the number '10'.
- Columns:** A text input field containing 'SummaryData'.
- Filters:** A section with a filter applied: 'Column: NamPartyTy' and 'Value: Person' with a close button (X).
- Results:** A list of search results. The first result is 'SummaryData', which is highlighted in yellow and has a mouse cursor over it.

It is possible to specify more than one *Results* field:

The screenshot shows the same search interface as above, but with multiple results specified:

- Columns:** A text input field containing 'SummaryData' with a close button (X) to its right.
- Results:** A list of search results. The first result is 'NamLast' and the second is 'NamFirst'. Both have close buttons (X) to their right.

In this example, rather than displaying *SummaryData*, both *NamFirst* and *NamLast* are specified:



Search	
wood	Go
3 records found.	
Wood	Angus
Wood	Angus
Wood	Gerard

Using a filter without including a Search box

It is possible to specify a filter without including a Search box in the Form:

The screenshot shows a search configuration window titled "Search". It contains several sections:


- Maximum Results:** An empty text input field.
- Columns:** A section with a "+ Add search column" button.
- Filters:** A section showing a filter configuration:
 - Column:
 - Value:
 - A close button (X) to the right.
- Results:** A section with a "+ Add results column" button.

When a user accesses the Form, a random record that meets the criteria specified by the filter will be displayed:

The screenshot shows a form titled "Add or edit Parties record" with a user profile "Gerard" in the top right. The form is divided into sections:

- Name:** A section containing several fields:
 - Party Type:
 - Title:
 - First Name:
 - Surname:
 - Other names: with a "+X" button.
 - with a "+X" button.
 - Birth Date:
 - Country: with a dropdown arrow.
 - Association: with a "+X" button.
- Image:** A section with a dropdown arrow.
- Biography:** A section with a dropdown arrow.

At the bottom of the form, there are navigation and action buttons: "<" and ">" arrows, "Record 1", a teal "Next" button, a "Save" button, and a "Reset" button.

Clicking  will search for and display another record at random, adding it to a record set and incrementing the record count on the left. Users can navigate back and forth through the record set using the two buttons:



Reloading the Form will reset the record set, displaying another randomly selected record and returning the record count to 1.



Note: Use the *Description* field in the Form Designer to include the Information icon in the Form and provide instructions / details about how users can use the Form.

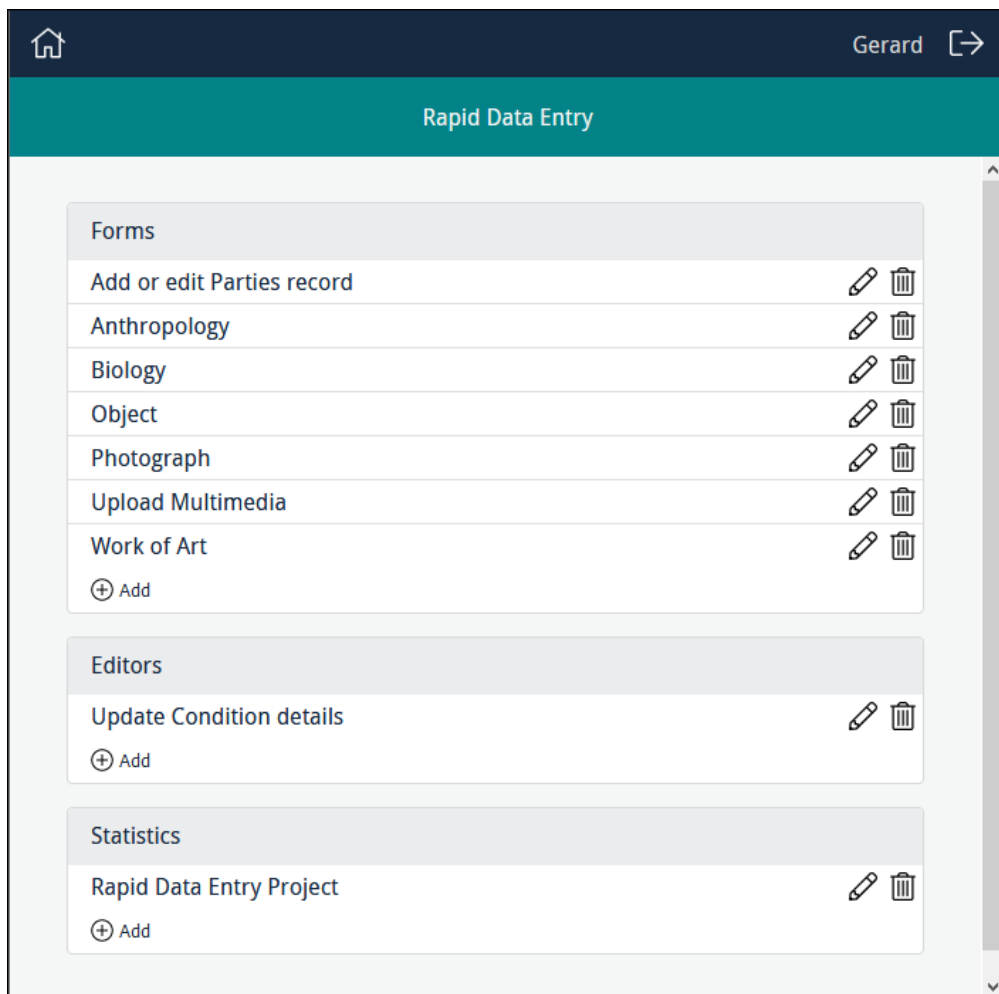
How to create and edit an Editor

A Sapphire Editor is, for all intents and purposes, a Global Replace tool. It enables authorised users to search for and select one or more records and:

- » Update a value in a field (e.g. perhaps replacing variations of a value with an approved value).
- » Attach the selected records to another record.

See *How to use an Editor to clean up data and process attachments* (page 22) for use cases.

When a Programme Lead accesses a project, any data entry Forms and Editors already created will display, along with an option to add a Form, Editor or Statistics tool:



1. To create a new Editor, click  in the Editors section:

2. Complete the top four fields:

<i>Title</i>	<p>Give the Editor a descriptive name.</p> <p>This name will be listed in the Editors section for authorised users.</p>
<i>Description</i>	<p>Add a description of the Editor or details about how to use it. When users access the Editor, an Information icon will display alongside the title:</p> <div style="border: 1px solid black; background-color: #008080; color: white; padding: 5px; display: inline-block; margin: 10px 0;"> Clean up Parties records i </div> <p>Clicking the icon will display a pop-up with the description.</p>

<i>Identifier</i>	In the <i>Identifier</i> field, add a descriptive identifier (or leave blank and an alphanumeric identifier will be assigned automatically). For the most part it is irrelevant whether the identifier is an alphanumeric value provided by the system or a label provided by the Programme Lead as it is only referenced behind the scenes by various components of Sapphire, or used to construct the URL for the Form. If you want a more human-readable URL, assign an identifier.
<i>Table</i>	The module that holds the records to be modified. Users will search for records in this module and select one or more records which will be updated by the Editor.

3. Specify permissions.

See *How to specify permissions* (page 89) for details.

4. Complete the *Source* field:

<i>Column</i>	The column that holds data which users will search in order to select records to be updated. For example, data entry users may have used the <i>Notes</i> field of the Catalogue module to store a range of details about objects, including an object's condition using keywords such as <i>Excellent</i> , <i>Poor</i> , etc. Setting the <i>Source</i> column to <i>NotNotes</i> will allow users to search <i>NotNotes</i> for the keywords and select matching records.
---------------	---

It is possible to filter values returned from *Column*.

i. Click . The button is rectangular with a light blue background and a dark border. It contains a blue plus sign in a circle followed by the text 'Add filter term' in a dark blue font.

ii. Specify a *Column* and *Value* to filter results displayed in the Source pane of the Editor.

Records displayed in the Source pane of the Editor must meet the criteria specified in *Filters*.

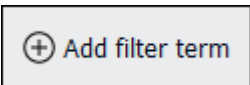
5. Complete the *Destination* fields.

The *Destination* is the field that will be updated by the Editor:

<i>Column</i>	The name of the field in <i>Table</i> specified above which will be updated by the Editor.
<i>Update only empty values</i>	This option operates in conjunction with <i>Column</i> . When the <i>Only Empty</i> checkbox is ticked, if the <i>Destination</i> field is already populated, it is not updated by this Editor.

Use when updating attachments	These two fields auto-complete if <i>Column</i> is an attachment field:	
	<i>Reference Table</i>	The target module. The attachment will be made to a record in this module.
	<i>Reference Column</i>	The data that will display in the <i>Target</i> pane of the Editor to assist users to identify and select the correct attachment record. By default, <i>SummaryData</i> is displayed. This can be changed to another field in <i>Reference Table</i> if necessary.
Use when replacing / cleaning values in a field	<i>Type</i>	Options are: » date The date picker pop-up will display when the cursor is placed in the field in the <i>Target</i> pane of the Editor. » text Values drawn from the <i>Lookup</i> specified will be auto-suggested as a value is keyed into the text box in the <i>Target</i> pane of the Editor.
	<i>Hint</i>	A hint to users. Text entered here will display in the text box in the <i>Target</i> pane of the Editor.
	<i>Lookup</i>	The name of a Lookup List. Values from this list will be listed in the <i>Target</i> pane of the Editor.
	<i>Level</i>	If the Lookup List is a multi-level hierarchy, use the <i>Level</i> field to specify which level of the hierarchy to display.

It is possible to filter values returned from Reference Column.

Click  to specify a *Column* and *Value* to filter results displayed in the *Target* pane of the Editor.

The screenshot shows the Sapphire Editor configuration window. At the top, the user 'Gerard' is logged in. The main configuration area includes:

- Title:** Clean up Object conditions
- Description:** Search the Notes field for condition information and update the Condition Status of selected records
- Identifier:** editor1304
- Table:** ecatalogue

Below these fields are several expandable sections:

- Permissions:** Shows 'Everyone' with a dropdown set to 'none'. An 'Additional Permissions' section contains one entry: 'group' (dropdown), 'Admin' (text), and 'use' (dropdown), with a close button (X). An 'Add permission' button is at the bottom.
- Source:** Shows 'Column' set to 'NotNotes'. A 'Filters' section with an 'Add filter term' button is below it.
- Destination:** Shows 'Column' set to 'ConConditionStatus'. It includes an 'Update only empty values' checkbox (unchecked). There are 'Reference' fields for 'Table' and 'Column', a 'Type' dropdown set to 'text', a 'Hint' field, a 'Lookup' field set to 'Condition Status', and a 'Level' field set to '1'. A 'Filters' section with an 'Add filter term' button is at the bottom.

At the bottom right, there are 'Save' and 'Reset' buttons.

6. Save the Editor.

Examples of Editors

This configuration is for an Editor that will update the *Condition Status* of records in the Catalogue module:

The screenshot shows the configuration interface for an editor. At the top, there is a navigation bar with a home icon, a folder icon, and the name 'Gerard' with a right-pointing arrow. Below this, the configuration is organized into several sections:

- Title:** Clean up Object conditions
- Description:** Search the Notes field for condition information and update the Condition Status of selected records
- Identifier:** editor1304
- Table:** ecatalogue

The **Permissions** section is expanded, showing:

- Everyone:** none
- Additional Permissions:** A list containing 'group' (dropdown), 'Admin' (text), 'use' (dropdown), and a close button (X).
- + Add permission** button.

The **Source** section is expanded, showing:

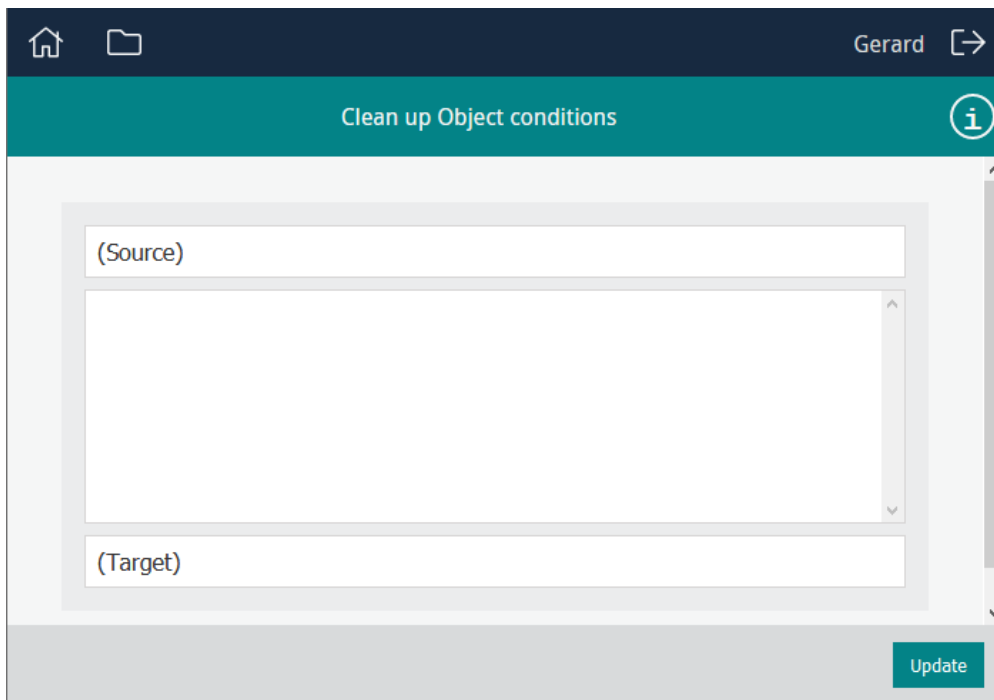
- Column:** NotNotes
- Filters:** + Add filter term

The **Destination** section is expanded, showing:

- Column:** ConConditionStatus
- Update only empty values:**
- Reference Table:** (empty)
- Reference Column:** (empty)
- Type:** text (dropdown)
- Hint:** (empty)
- Lookup:** Condition Status
- Level:** 1
- Filters:** + Add filter term

At the bottom right, there are two buttons: **Save** and **Reset**.

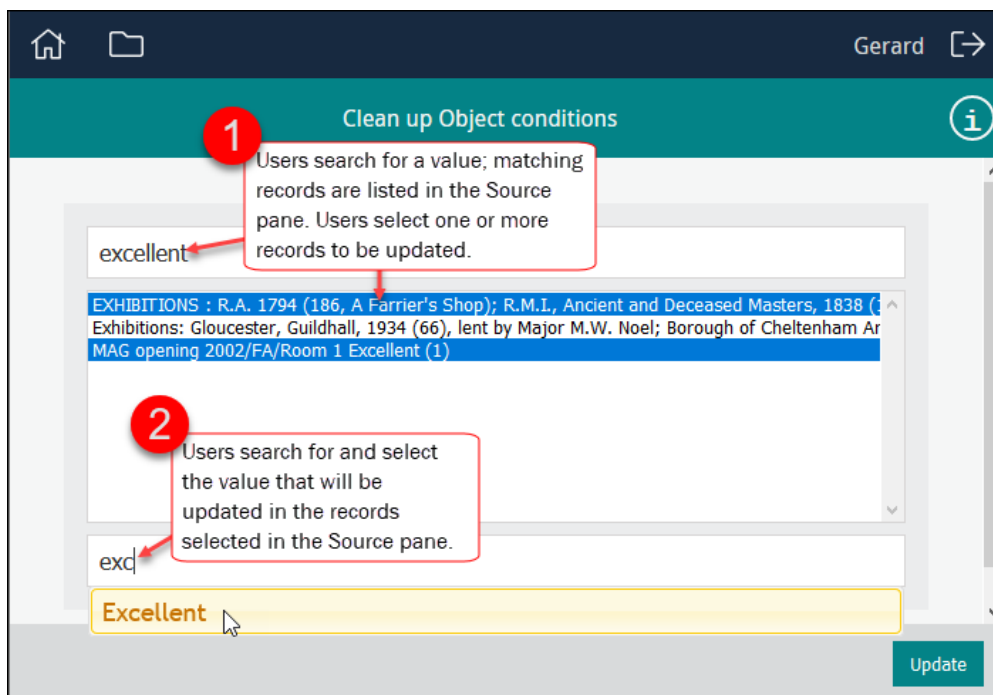
It will display for users as:



Authorised users (only members of group Admin in this example):

1. Key a search term in *(Source)*.
A search for the term, *excellent* in this example, will be performed in the *Notes* field. Any matching records will be listed in the *Source* pane.
2. Select one or more matching records.
3. Begin keying a value into the *(Target)* field and select the required value.

In this example, values in the *Condition Status* Lookup List will be displayed:



4. Click **Update** to update the selected records.

This configuration is for an Editor that will attach records in the Catalogue module to a Parties record for the person who performed a Condition Check:

The screenshot shows the configuration interface for an Editor in the Sapphire system. The interface is titled "Gerard" and displays the following configuration details:

- Title:** Update Condition details
- Description:** Search the Notes field for details of the person who performed a Condition check and attach selected records to the appropriate Parties record
- Identifier:** editor1286
- Table:** ecatalogue
- Permissions:**
 - Everyone: none
 - Additional Permissions: group Admin use
 - + Add permission
- Source:**
 - Column: NotNotes
 - Filters: + Add filter term
- Destination:**
 - Column: ConCheckedByRef
 - Update only empty values:
 - Reference Table: eparties
 - Reference Column: SummaryData
 - Type: text
 - Hint:
 - Lookup:
 - Level:
 - Filters:
 - Column: NamPartyTyp
 - Value: Person

At the bottom right, there are "Save" and "Reset" buttons.

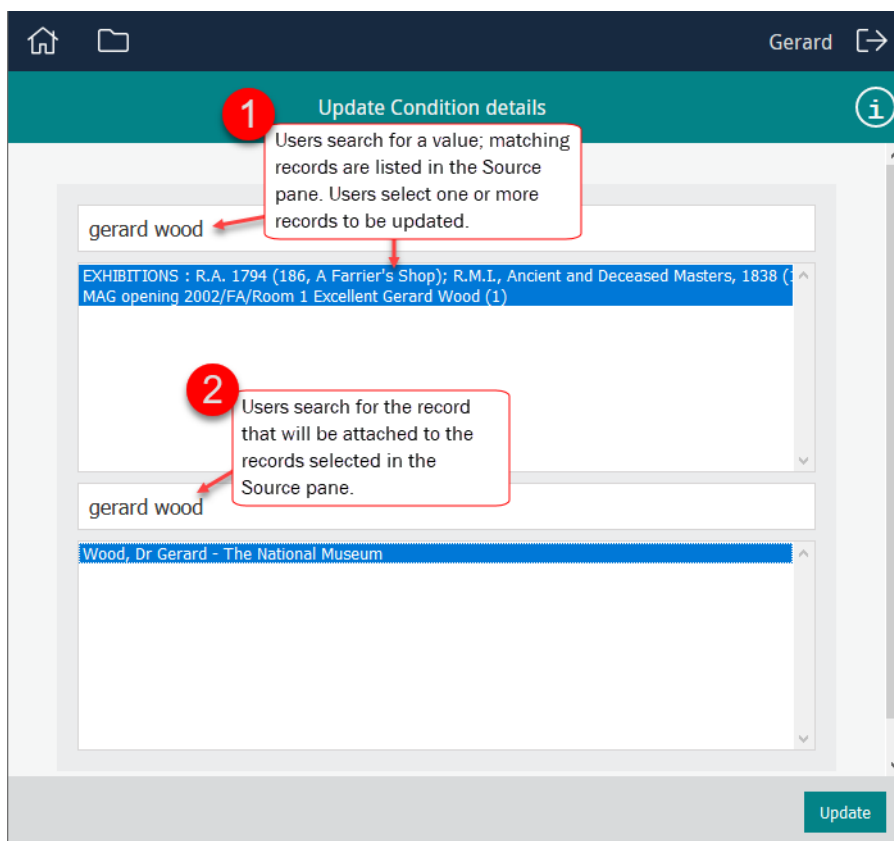
It will display for users as:

The screenshot shows a web application interface for updating condition details. The top navigation bar is dark blue and contains a home icon, a folder icon, the user name 'Gerard', and a refresh icon. Below this is a teal header bar with the text 'Update Condition details' and an information icon. The main content area is a light gray box containing two input fields: '(Source)' and '(Target)'. Each input field is followed by a large, empty, scrollable list box. At the bottom right of the form is a teal 'Update' button.

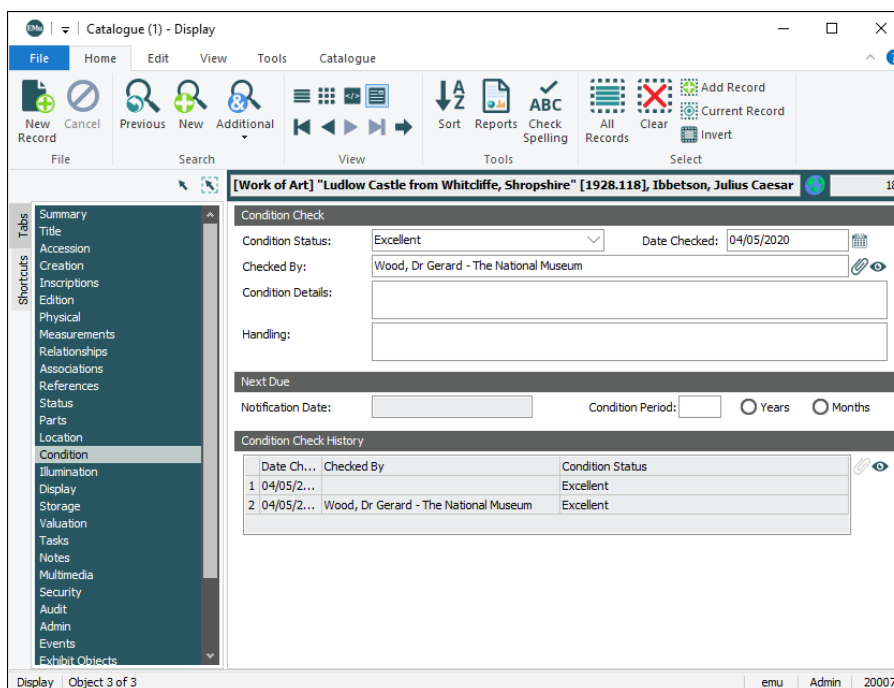
Authorised users (only members of group `Admin` in this example):

1. Key a search term in *(Source)*.
In this example, a user would enter a name and a search will be performed in the *Notes* field. Any matching records will be listed in the *Source* pane.
2. Select one or more matching records.
3. Begin keying a value into the *(Target)* field and select the required value.
In this case a search will be performed in the *Parties* module.

4. Select the required Parties record:

5. Click **Update** to update the selected records.

The selected Catalogue records will be attached to the selected Parties record using the *Checked By: (Condition Check)* field:

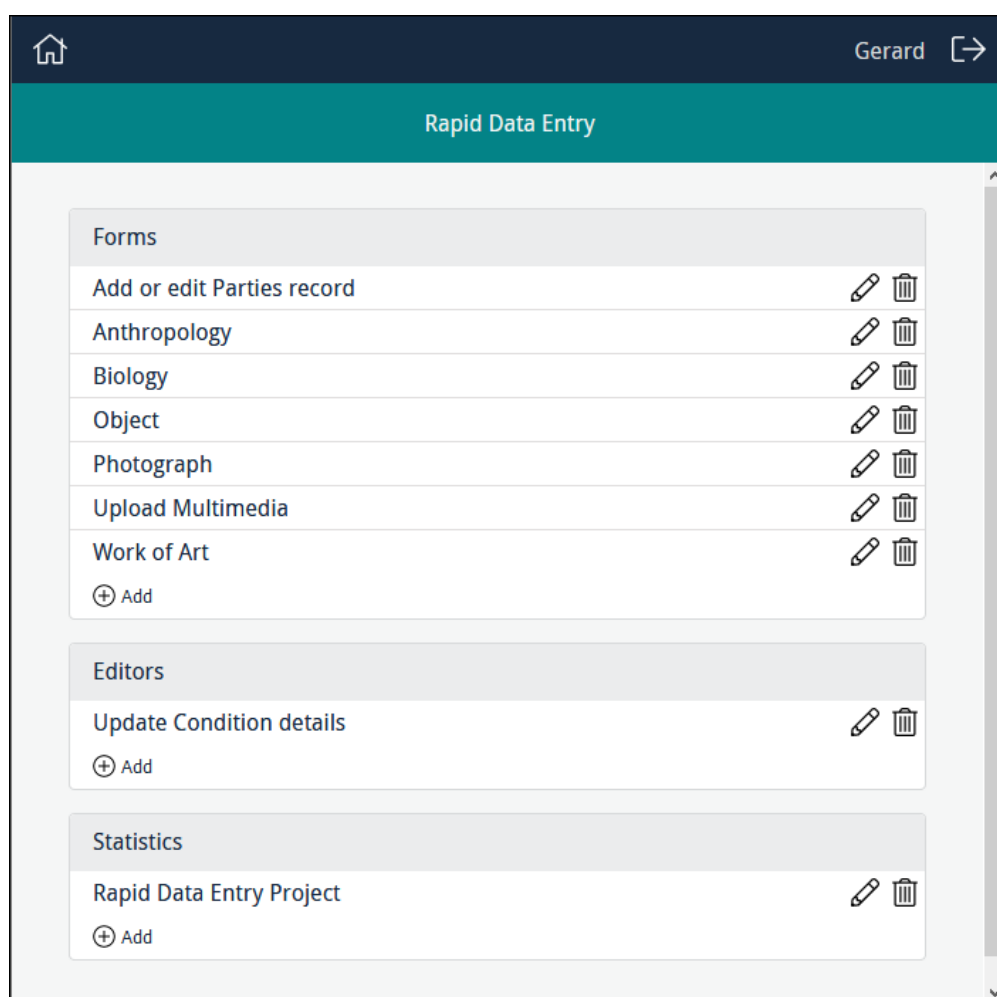



How to display Statistics

Authorised users (Managers) can be provided with graphs (bar and pie) presenting any sort of statistical data (the progress of a project, the condition of objects, etc.). A graph can be pointed at any column in any module, but some columns will be more useful to represent in a graph than others. This example graphically displays the condition of objects in the Catalogue:



Available graphs (bar and pie) are listed under the Statistics heading:



1. To create a new graph, click  in the Statistics section.
2. Complete the top four fields:

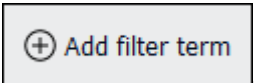
<i>Title</i>	Give the graph a descriptive name. This name will be listed in the Statistics section for authorised users.
<i>Description</i>	Add a description of the purpose of the graph. When users access the graph, an Information icon will display alongside the title. Clicking the icon will display a pop-up with details about the graph.
<i>Identifier</i>	In the <i>Identifier</i> field, add a descriptive identifier (or leave blank and an alphanumeric identifier will be assigned automatically).
<i>Table</i>	The module that holds the data that will be graphed.

3. Specify permissions.
See *How to specify permissions* (page 89) for details.

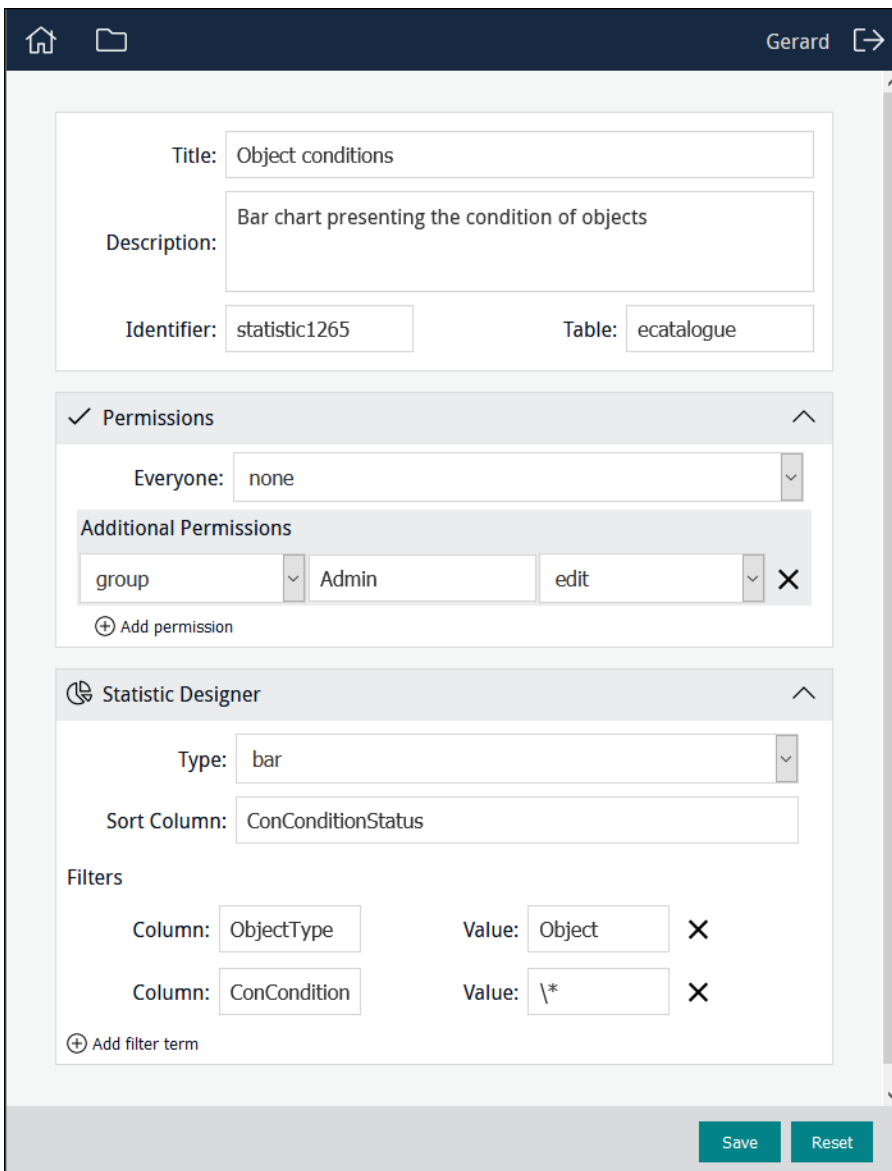
4. Complete the *Statistic Designer* fields:

Type	The type of graph: >> bar >> pie
Sort Column	The column holding the data to be graphed.

5. It is possible to filter values returned from *Sort Column*:

- i. Click .
- ii. Specify a *Column* and *Value* to filter results displayed in the graph.

The graph shown above is configured as:



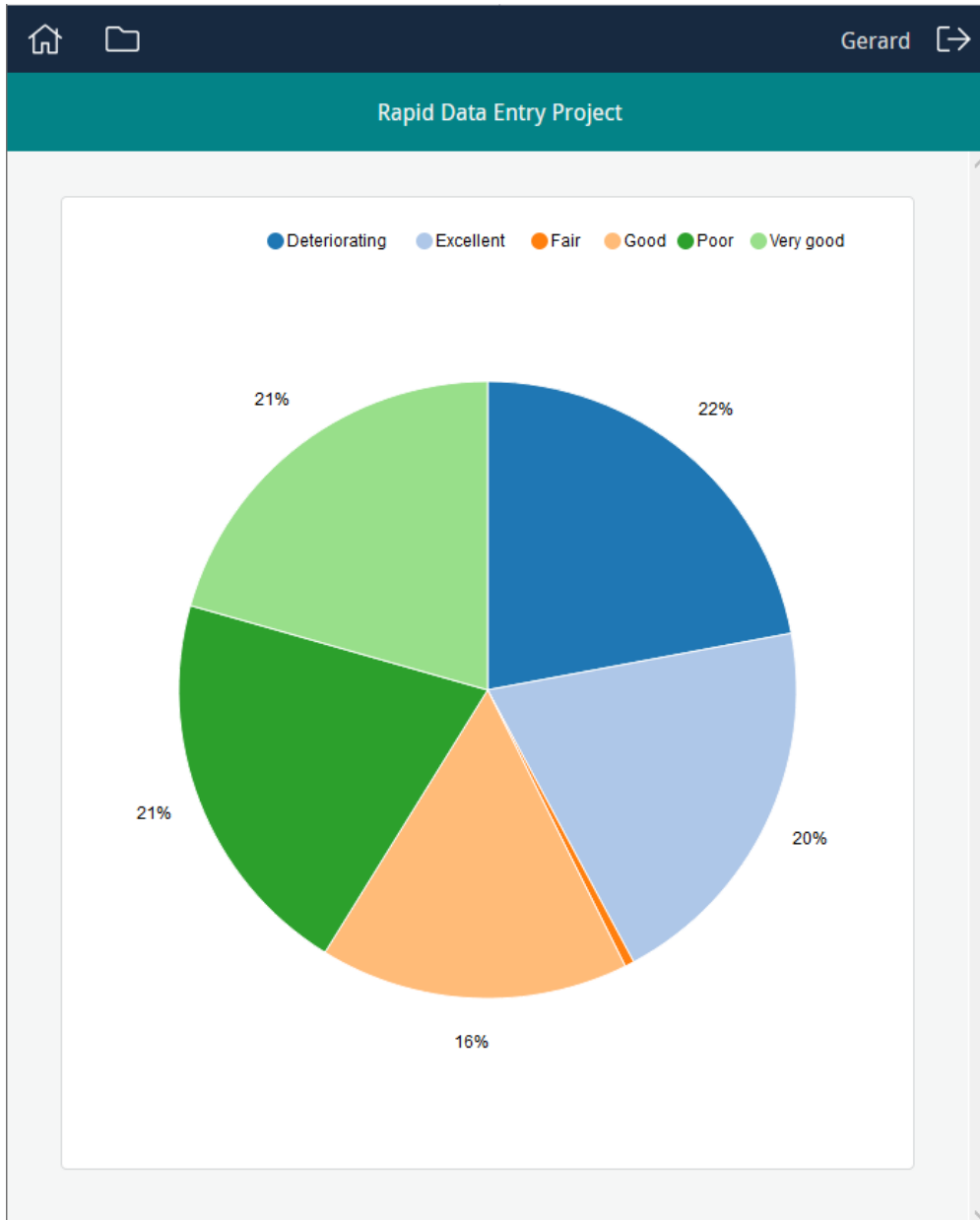
6. Save the graph.

Manager

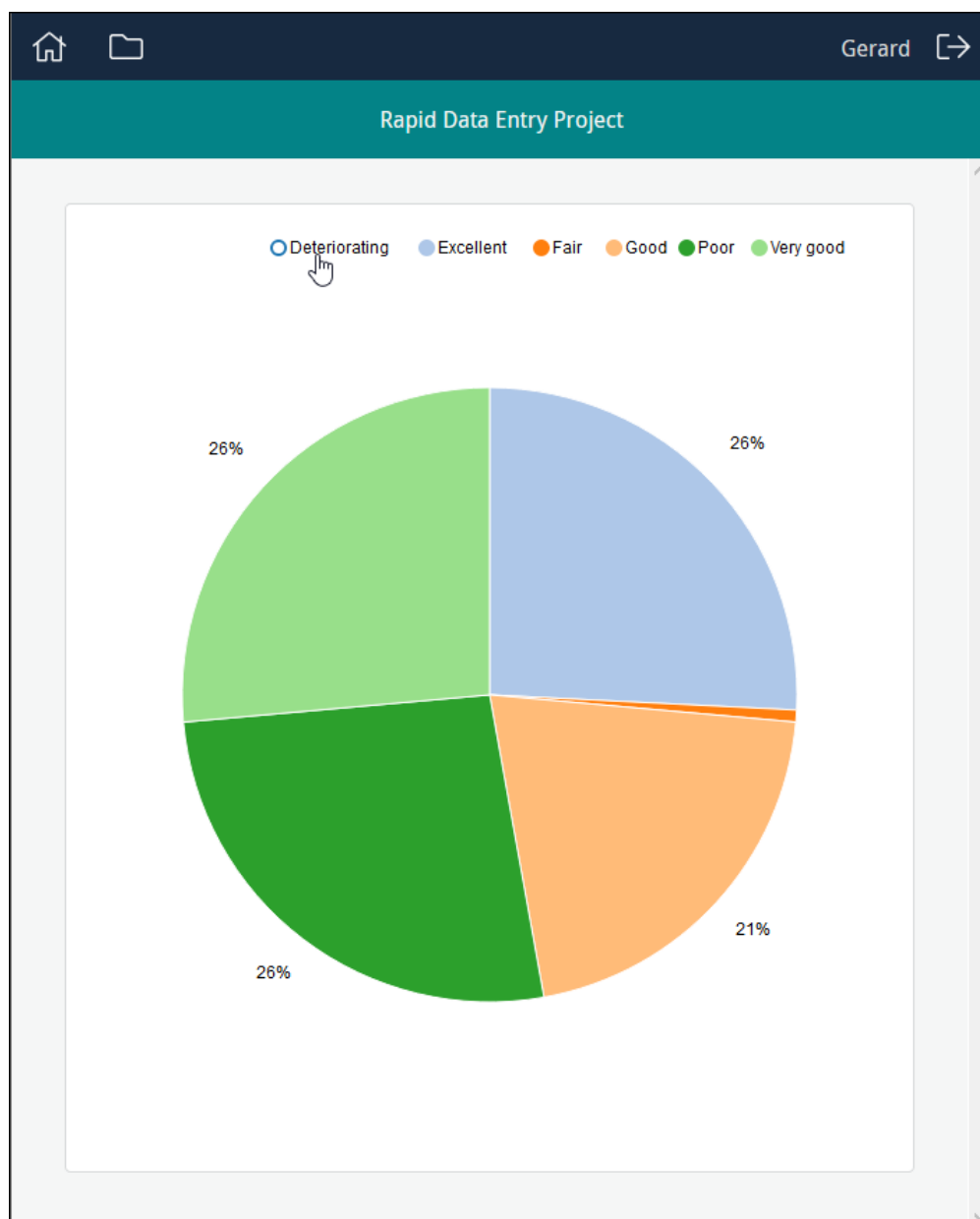
Authorised users (Managers) can be provided with graphs (bar and pie) presenting any sort of statistical data. (the progress of a project, the condition of objects, etc.). This example presents the condition of objects as a bar graph:



The pie chart depicts the same results as a percentage:



The pie chart is interactive. Clicking an entry in the Legend above the chart (**Deteriorating** in this example), removes the entry from the graph and recalculates the percentages of the remaining entries:



Minimum System Requirements

EMu 5.1 (1710021) is the minimum version of EMu that runs with Sapphire 1.2.0.

Sapphire is designed to operate in browsers which support a subset of HTML5. This includes:

- » Desktop environments running fairly recent versions of most major browsers:
 - » Internet Explorer version 10 or above
 - » Firefox version 4 or above
 - » Chrome version 13 or above
 - » Safari version 6 or above
- » iOS devices running iOS 6 or above
- » Android devices running:
 - » Android browser version 4.4 or above
 - » Chrome version 36 or above
 - » Firefox version 31 or above



Note: The Sapphire tools will often operate successfully with earlier versions of these browsers. However, testing has shown that not all functionality is available in these browsers. For example, while many Forms will operate in Internet Explorer 9, this version of IE does not support the HTML5 function that enables a user to select files when attaching images.

Release Notes

What's New in Sapphire 1.2.0:

- » Restyled skin to match Axiell's branding.
- » Updated button labels to eliminate confusing / misleading text.
- » New warnings / prompts before continuing certain actions.
- » Redesigned page layout providing better access to commonly used features.
- » Responsive Form layout: multi-column forms respond when viewed on narrow screens.
- » User group selection at login.
- » Improved UI.
- » New Form control enabling upload of all multimedia types.
- » Improved multi-value controls: rows can be added / removed from anywhere in a multi-value field.
- » Drop-in background: Sapphire can be re-branded by dropping in an image to use as background.

Bug Fixes

- » Form data would be refreshed automatically after an error while saving.
- » Button icons rendering strangely after recent Chrome update.
- » Incompatible date format configurations between UI and Server causes display / save errors.
- » Date-picker and Image controls clash.
- » In some time zones date values are stored as previous day.
- » Firefox displays inconsistent fonts in Form.
- » Form designer description input field overlaps other input fields in Firefox.
- » Ref Table / Column configuration fields in Form designer do not clear after changing field type from attachment.
- » Image control stores images in wrong order.
- » Tabbing between fields does not occur sequentially.
- » Form designer does not auto hide line configuration.
- » Forms appear blank if custom layer added and then removed.
- » Image control buttons not responding on mobile devices.
- » Form designer auto suggests fat client only columns - fixed in EMu 6.2 (2004241).
- » Form search result count displays n + 1 results - fixed in EMu 6.2 (2004031).

INDEX

A

Attachment field 9, 67

C

Create attachments 28
Create Editor 98
Create Project 39
Customise form layout 83

D

Data capture 1
Data cleansing 1, 22
Data Entry user 7
Date field 9
Deferral system 82

E

Edit a record 20
Editor 22

G

Graphs 108

I

Images 17

K

Kind
 image 71
 image-capture 79
 mobile-image-capture 81
 multiple 57

L

Logging in / out 5, 37
Lookup 65

M

Manager 111
Minimum System Requirements 115
Multimedia 12, 14
Multimedia fields 70

O

Overview 1

P

Permissions 89
Polygons 15
Programme Lead 32, 37

S

Search box 91
Search for a record 20
Select image 14
Statistics 108

T

Type
 date 59
 selection 61
 text 63

U

Upgrade 3

V

Version 3