

QGIS: Working with Widgets – Part 1

In a recent Cadline FAQ I documented how to use **Widgets** within QGIS to automatically add a **unique ID** to a record once the user has finished digitising the feature:

<https://www.cadlinecommunity.co.uk/hc/en-us/articles/360000030557-QGIS-Widgets-Auto-Add-Unique-ID>

In this White Paper we will now explore several other widgets to increase your automation when working within QGIS, these options include:

- Working with **Unique ID** values
- Provide users with a **Value Map** providing a list of drop down options when entering attribute values
- Automatically updating a **Date** field
- Automatically updating an **AREA** value for features

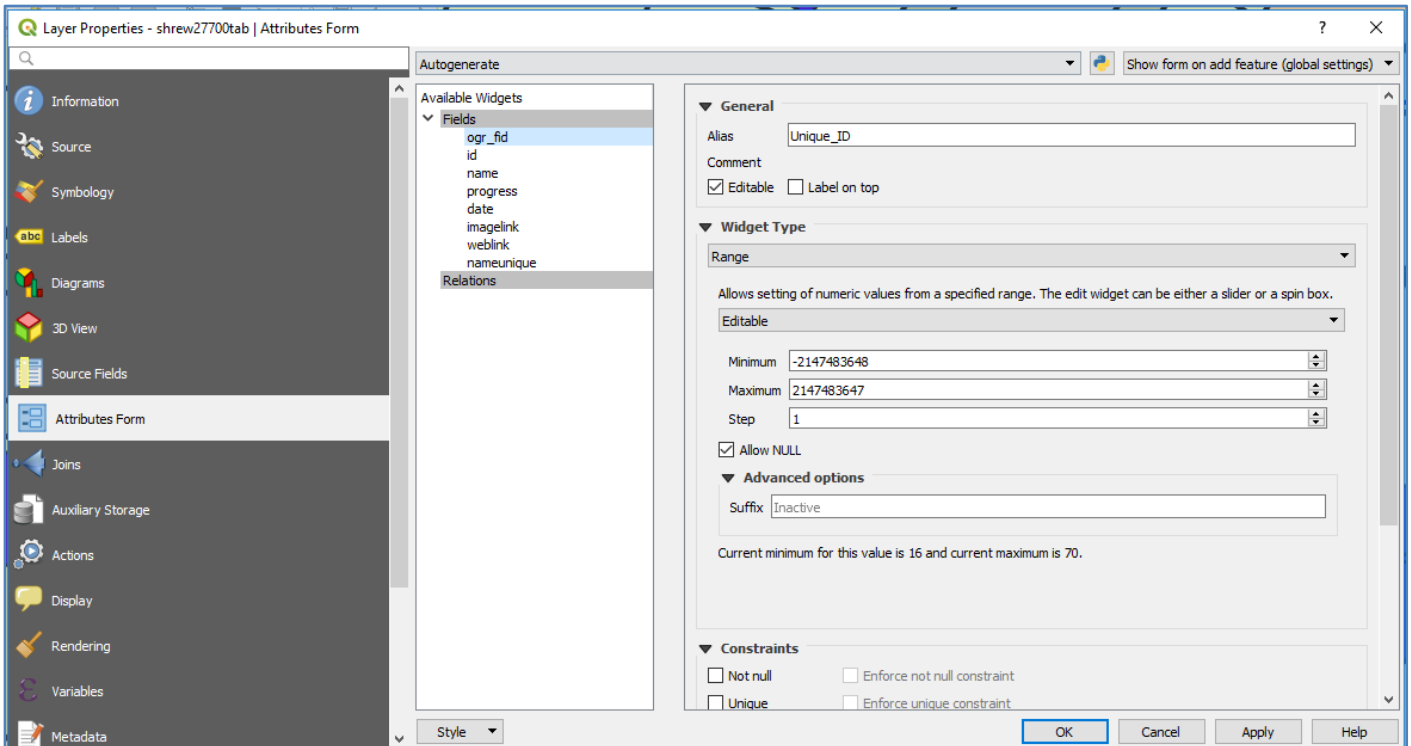
(1) Working with Unique Values

In this section we will explore the best options for working with unique values for your spatial records. This includes options to configure if you reveal unique values to the user, and options for automatically creating unique values for new records.

1a – Using an ALIAS:

Within QGIS (version 3 onwards) the **Widgets** for configuring your layer attributes are stored in the Layer Properties > **Attribute Form**.





Once within the Attributes Form window, firstly choose the Field that you wish to configure. In this case we will choose the **ogr_fid** field.

Using the **Alias** option, you can choose to provide the user with an **alternative field name** within both the Attribute window and in the Edit Attributes form (used when they are creating a new feature). If we change the value to be called **Unique_ID**, then press **OK** the change will be saved within the project.

shrew27700tab :: Features Total: 38, Filtered: 38, Selected: 0

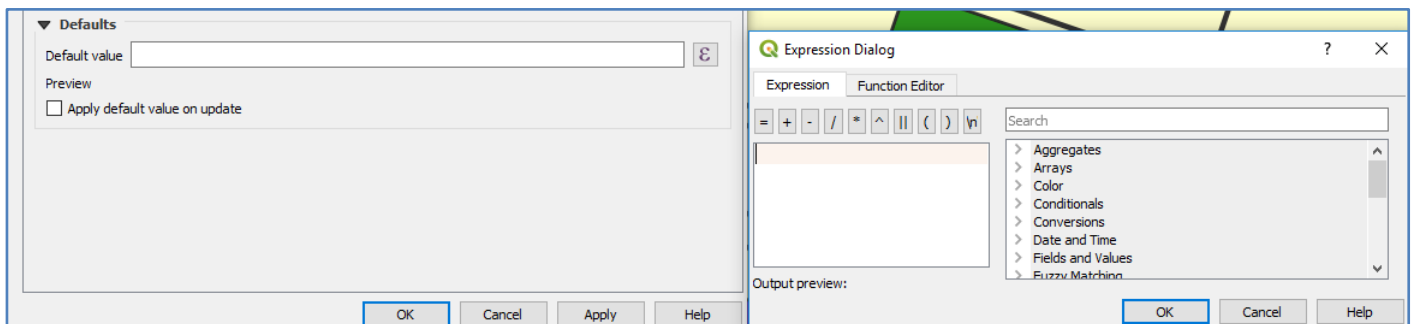
	Unique_ID	id	name	progress	date
1	16		Leicester	Passed	2018-06-08
2	17		leicester	Submitted	2018-04-12
3	18		leicester	Declined	NULL
4	21		leicester	Declined	NULL
5	22		leicester	Passed	NULL

1b – Applying a Default Value:

In addition, we can use the **Text Edit Widget** to automatically create a **default value** for the Feature/Unique ID value as we create new records into our spatial datasets. These Widgets are available within the **Attributes Form** of the **Layer Properties**.

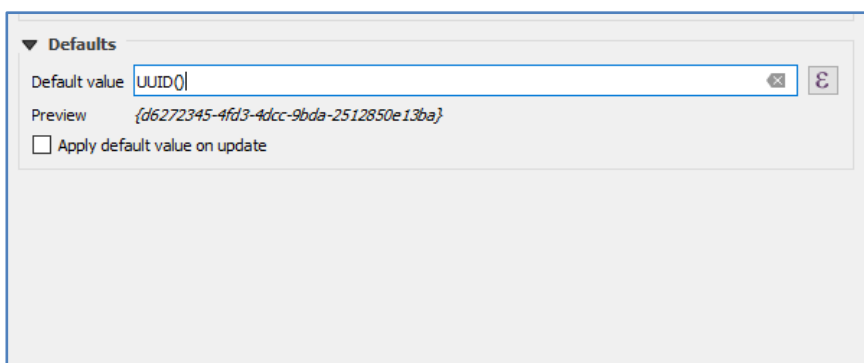
Within the **Attributes Form**, firstly select the Field that you wish to automatically create, in this case we will choose the **ogr_fid**.

Having chosen the **ogr_fid** Field, within the right-hand pane choose the **Widget Type** to be **Text Edit** and then in the **Defaults** section choose the **Expression icon** to type an expression to use for the default.



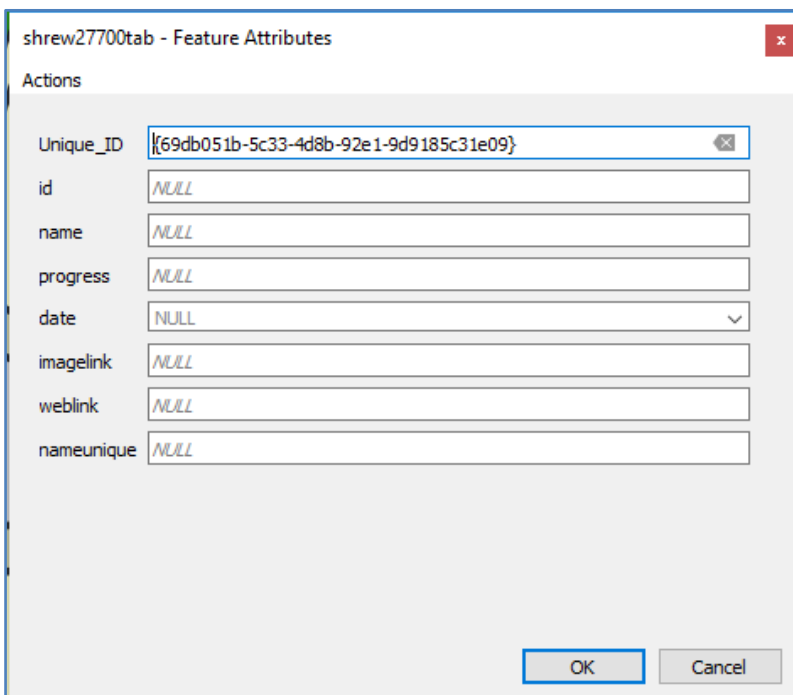
Add the expression of **UUID()**

OK the Expression Dialog, and the **Default Value** of a Unique ID will be added each time a new record is created, with a preview of what that value will look like.



Note – that you can also re-apply a new UUID value if the user updates the record.

Apply the changes to the **Layer Properties** and now create a new feature. Once you have digitised the new feature the **Attributes Form** will auto open and the Unique ID field will be auto populated. Finish the attribute data entry and **Save** the changes to your data.



shrew27700tab - Feature Attributes

Actions

Unique_ID	{69db051b-5c33-4d8b-92e1-9d9185c31e09}
id	NULL
name	NULL
progress	NULL
date	NULL
imagelink	NULL
weblink	NULL
nameunique	NULL

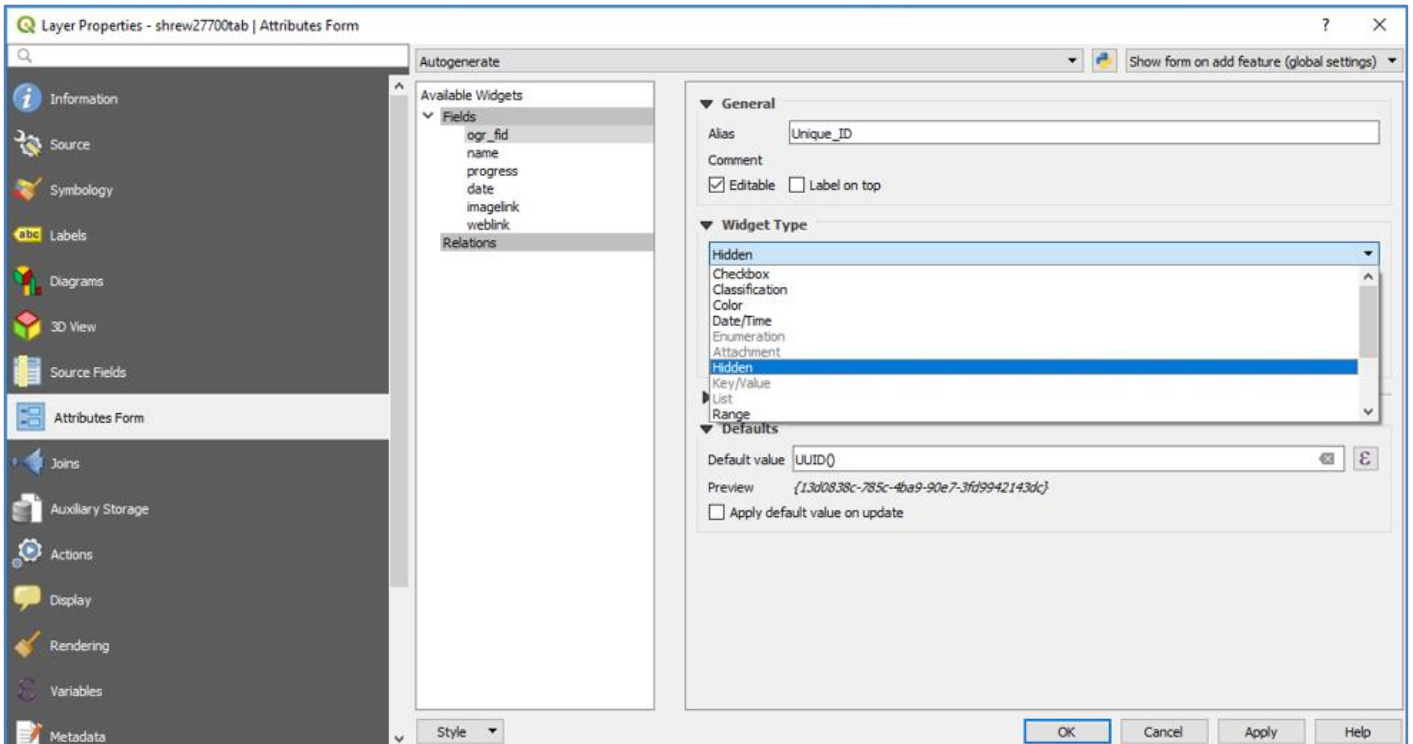
OK Cancel

1c - Hiding Unique ID Values:

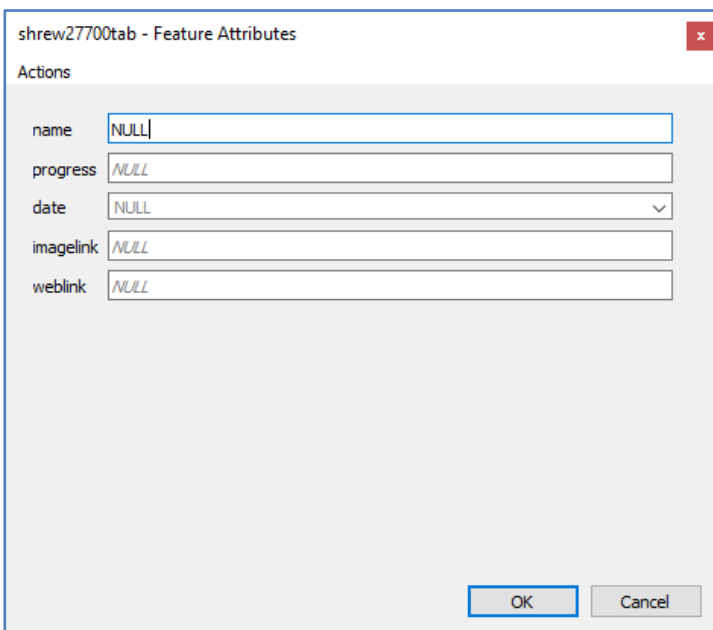
In many instances while you need the Unique_ID value to be updated and created, you don't necessarily need to show the user these values. You can therefore use the Layer Properties to **hide** any such values.

Within the **Attributes Form**, firstly select the Field that you wish to hide, in this case we will choose the **ogr_fid**.

Having chosen the ogr_fid Field, within the right-hand pane choose the **Widget Type** to be **Hidden** and press **OK**.



If the user now creates a new feature, the **Edit Attributes form** will now no longer display the Unique_ID field. This means you can choose which fields the user has the options to see and therefore edit.



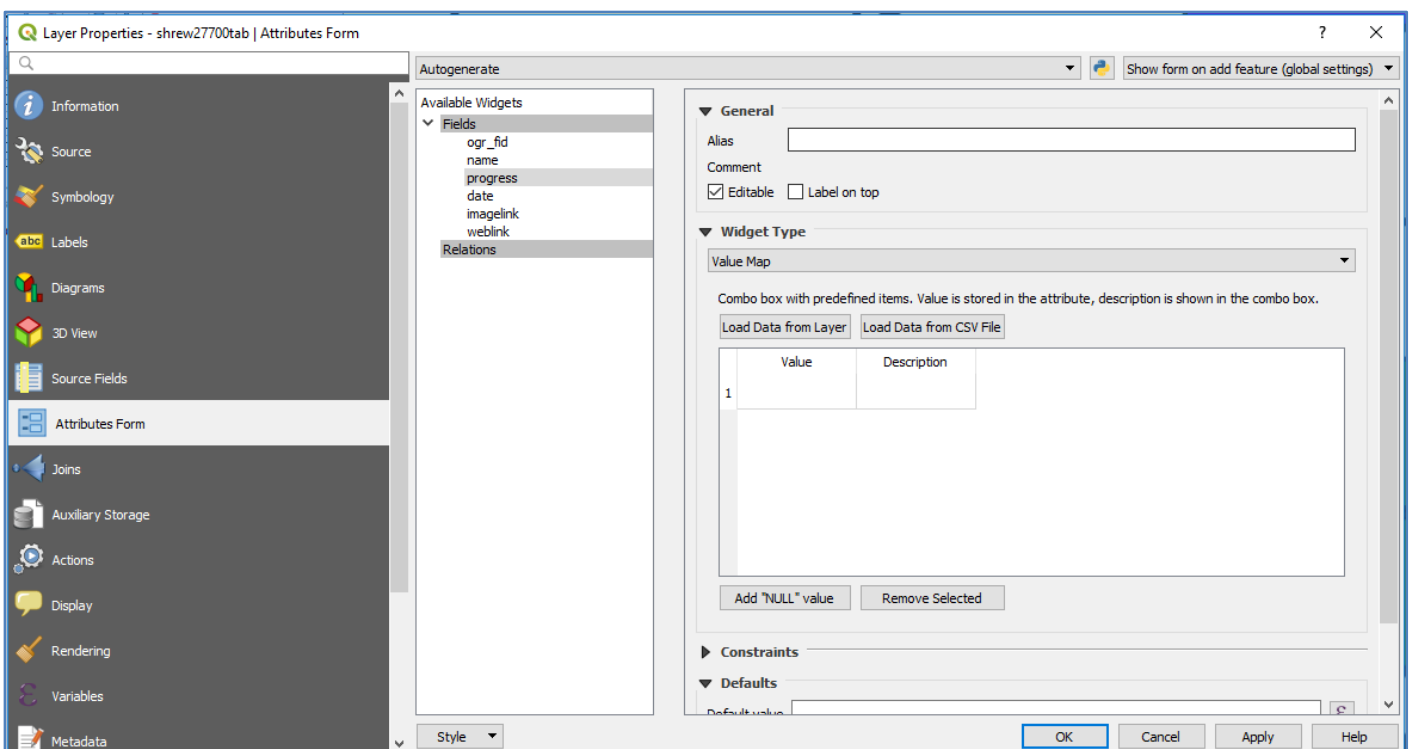
(2) Working with Value Maps

In this section we will explore the best options for working with **Values Maps**, which provide a List of values for your spatial records. Creating a Value Map allows you to control what Attribute Values a user can choose, and therefore enables you to minimise errors in your attribute data. It can be commonly used for well-known values such as crime types or the progress of a planning application.

2a – Creating a Value Map:

Within QGIS (version 3 onwards) the **Widgets** for configuring your layer attributes are stored in the Layer Properties > **Attribute Form**.

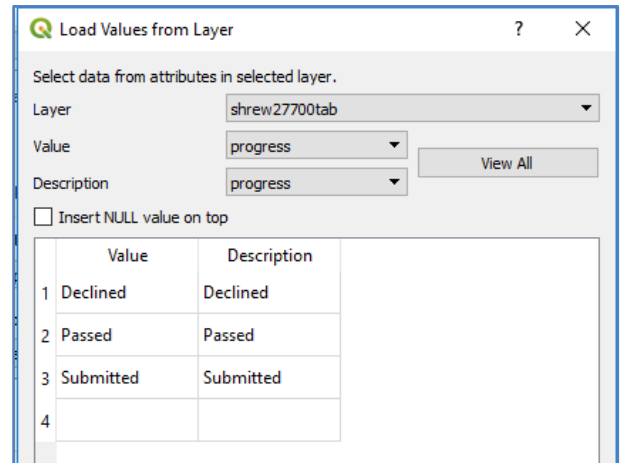
Once within the Attributes Form window, firstly choose the Field that you wish to configure. In this case we will choose the **progress** field, and then choose the Widget to be **Value Map**.



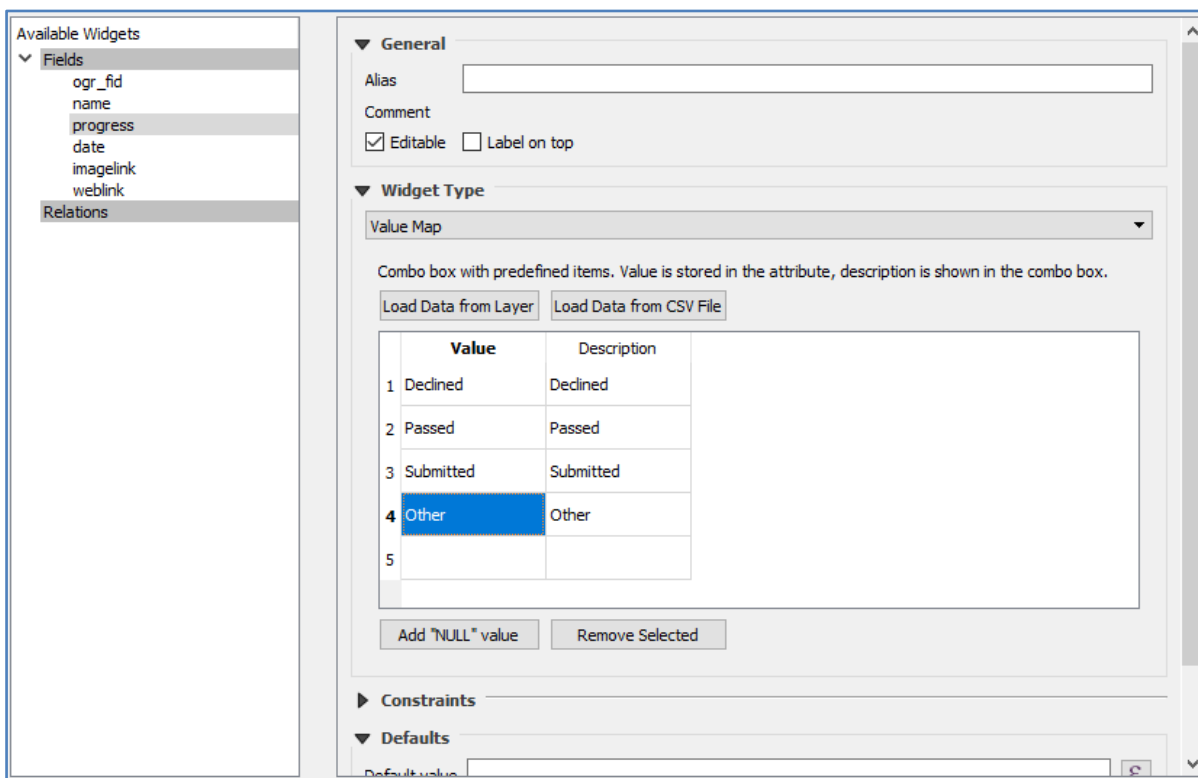
Using the **Value Map** option, you can create a list of values that the user can choose from. The list can be created by manually typing a value, loading from a CSV or loading the current unique values from within the layer. In this example we will choose the option - **Load Data from Layer**.

In the Load Values from Layer window, choose the following options:

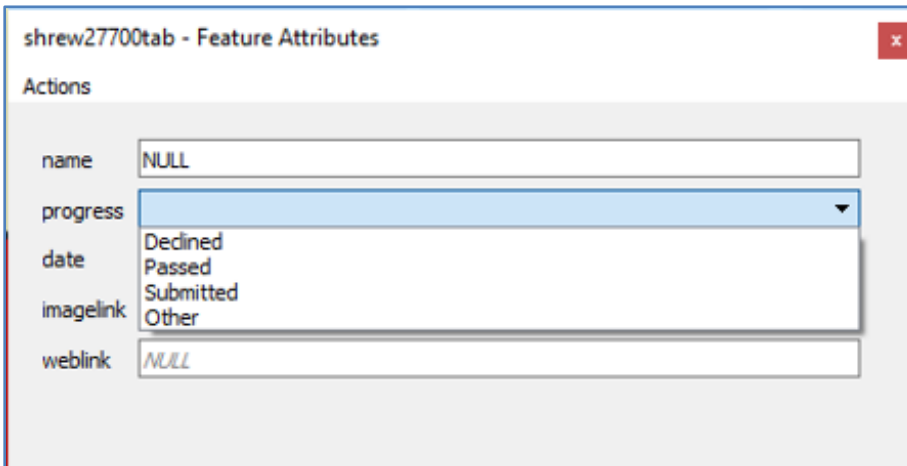
- **Layer** = Planning Apps Layer (Shrew27700tab)
- **Value** = the Progress field
- **Description** = the Progress field



If required, you can also add **manual List** choices e.g. **Other**.



Now when the user creates a new feature, the **Edit Attributes** form will provide a list box with the 4 unique values for the user to choose from. In this way you can ensure that users enter the values which are valid, and negates misspellings etc.



shrew27700tab - Feature Attributes

Actions

name: NULL

progress: Declined, Passed, Submitted, Other

date:

imagelink:

weblink: NULL

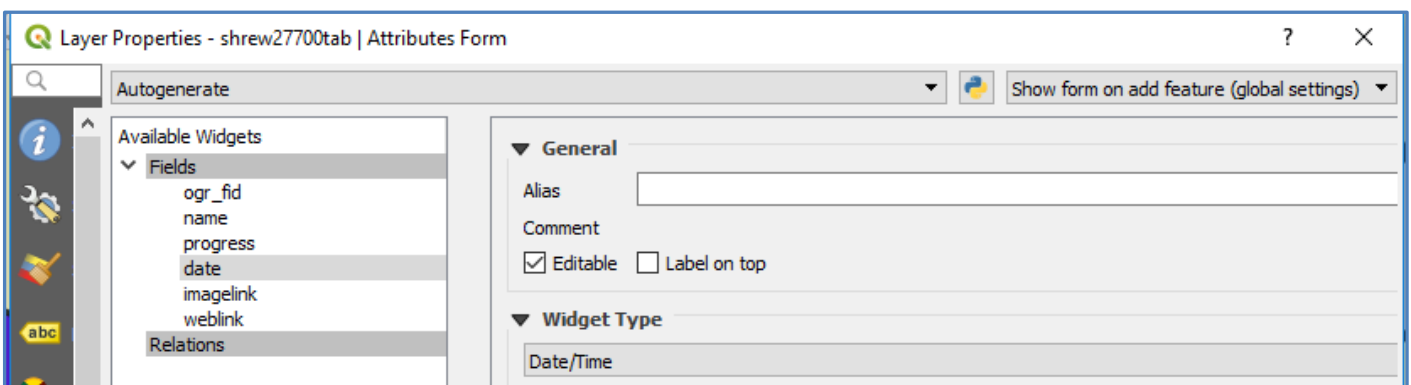
(3) Working with Dates

In this section we will explore the best options for working with **Date Values**. QGIS allows you to either manually type a date value, which can lead to incorrectly formatted values and missing dates or using the Widget tool you can provide users with a **data picker** to enable them to more easily choose date values.

3a – Creating a Date Picker:

Within QGIS (version 3 onwards) the **Widgets** for configuring your layer attributes are stored in the Layer Properties > **Attribute Form**.

Once within the Attributes Form window, firstly choose the Field that you wish to configure. In this case we will choose the **date** field, and then choose the Widget to be **Date/Time**.



Layer Properties - shrew27700tab | Attributes Form

Autogenerate

Show form on add feature (global settings)

Available Widgets

- Fields
 - ogr_fid
 - name
 - progress
 - date
 - imagelink
 - weblink
- Relations

General

Alias

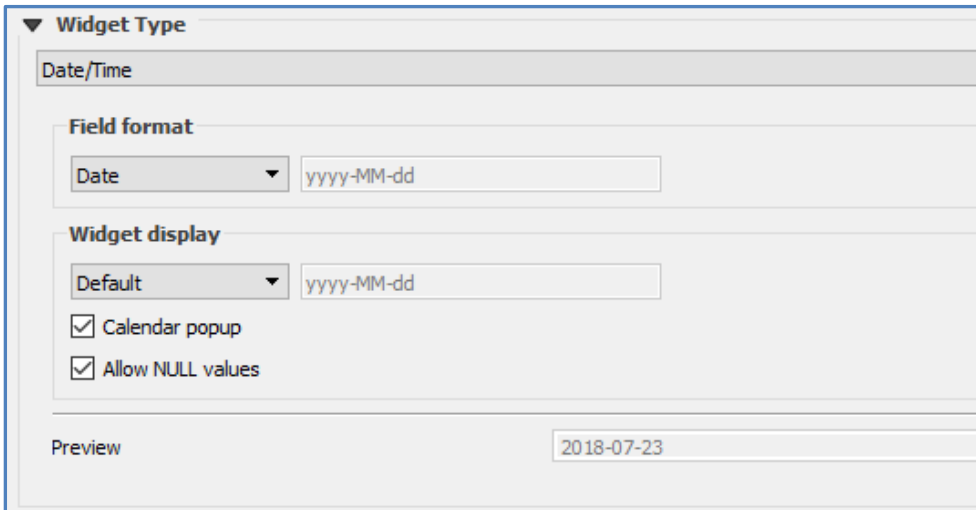
Comment

Editable Label on top

Widget Type

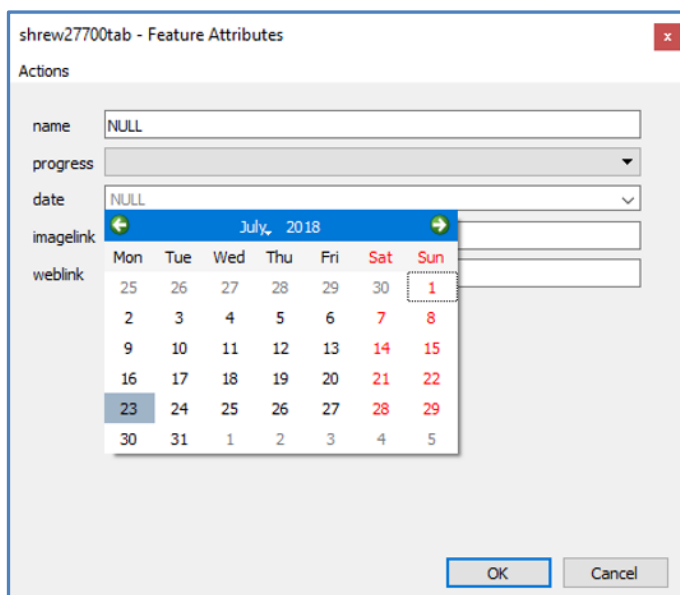
Date/Time

Using the **Date/Time** option, you can choose the field format, which could be Date or Date and Time, as well as choose the format for the Date (and time). In addition, you can reveal a calendar popup so that users can easily pick a suitable date. In this example we will choose the **Date** option and choose to reveal the **Calendar Popup**.



The screenshot shows the 'Widget Type' configuration dialog for a 'Date/Time' widget. Under 'Field format', the 'Date' dropdown is selected with the format 'yyyy-MM-dd'. Under 'Widget display', the 'Default' dropdown is selected with the format 'yyyy-MM-dd'. Two checkboxes are checked: 'Calendar popup' and 'Allow NULL values'. A 'Preview' field at the bottom shows the date '2018-07-23'.

Now when the user creates a new feature, the **Edit Attributes form** will provide a **calendar popup** where the user either chooses today's date, or can scroll backwards and forwards as required.



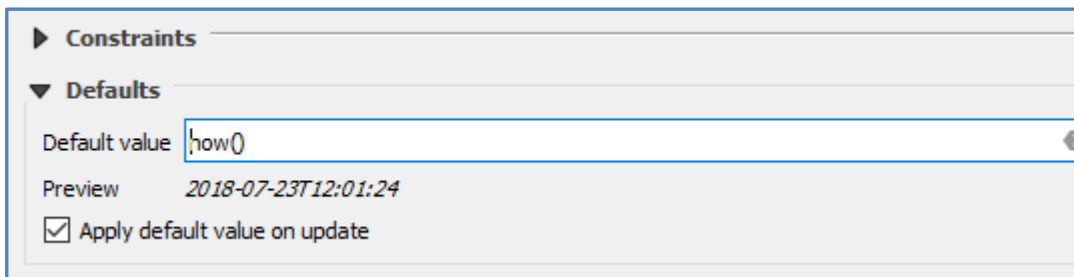
The screenshot shows the 'shrew27700tab - Feature Attributes' dialog. The 'date' field is set to 'NULL' and a calendar popup is displayed for July 2018. The calendar shows the days of the week (Mon-Sun) and the dates from 1 to 31. The date '23' is highlighted in blue. The 'date' field in the dialog is currently set to 'NULL'.

3b – Apply a Default Date:

In addition, we can update the **Widget** to automatically create a **default value** for the Date value. We can also choose to Apply this default whenever a record is updated. This can be useful to allow you to track when changes were made to your datasets.

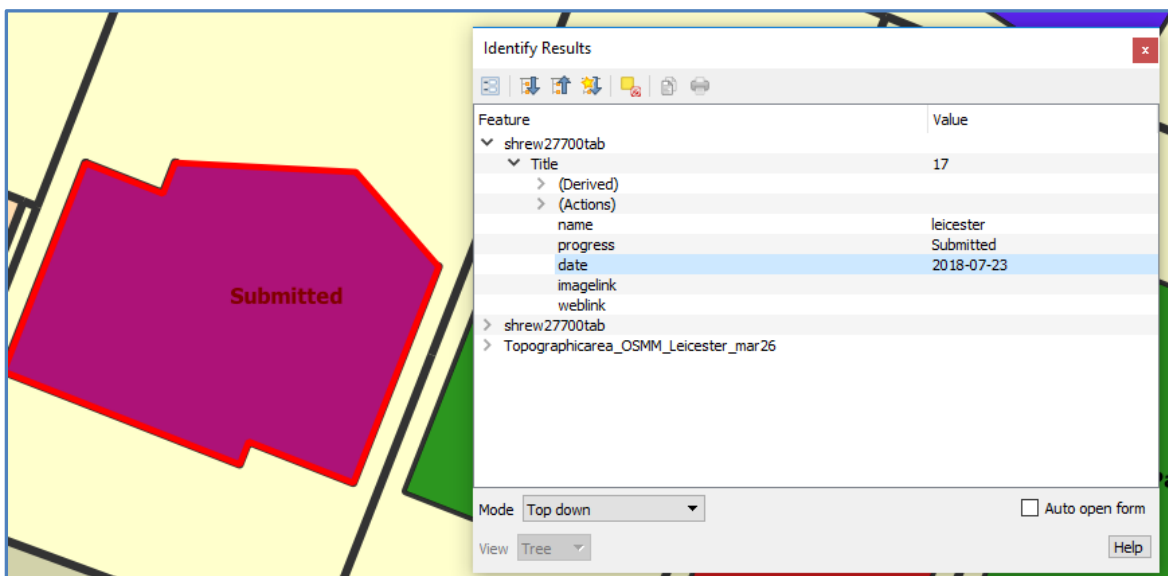
Within the **Attributes Form**, firstly select the Field that you wish to automatically create, in this case we will choose the **Date** field.

Having chosen the Date Field, within the right-hand pane scroll down to the **Default options** and type a Default Value of **Now()** and notice the Preview updates to show today's date and time.



Ensure you tick to **Apply the default value on update** and OK the Layer Properties.

If you now edit an existing feature, e.g. by **reshaping** it, notice that when you save the layer, the Date field will auto update to show today's date.



(4) Auto Calculating Area Values

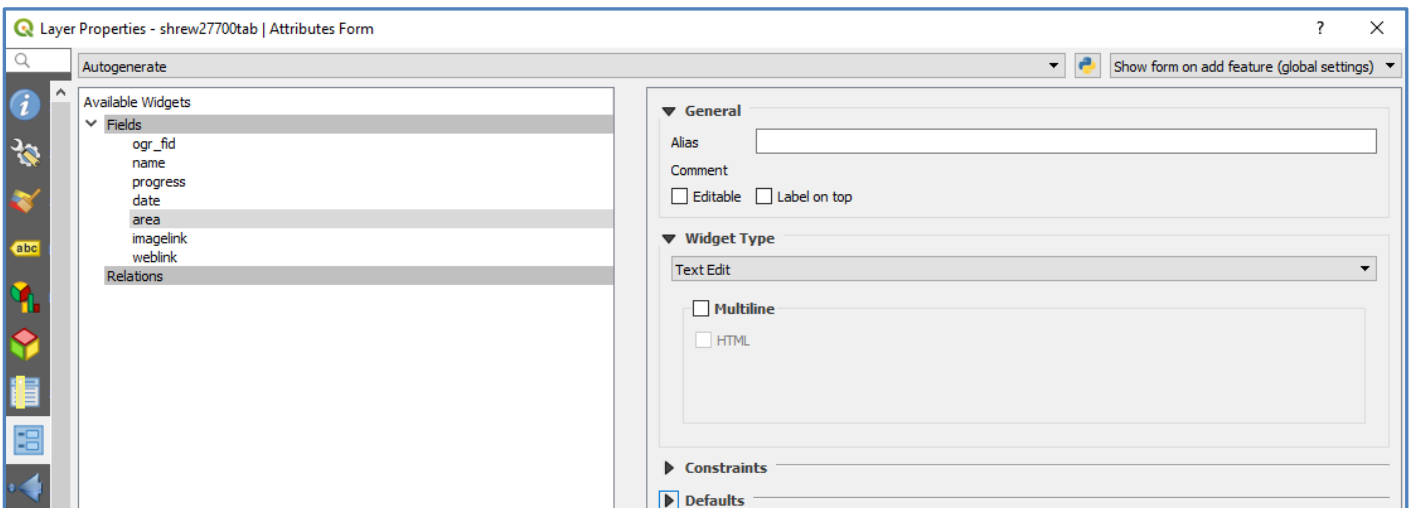
In this final section we will explore the best options for auto calculating the **Area** of objects within QGIS. This can be useful as it means you can reveal geometric values to your users without needing to extract these values manually.

4a – Calculating Area Values

Within QGIS (version 3 onwards) the **Widgets** for configuring your layer attributes are stored in the Layer Properties > **Attribute Form**.

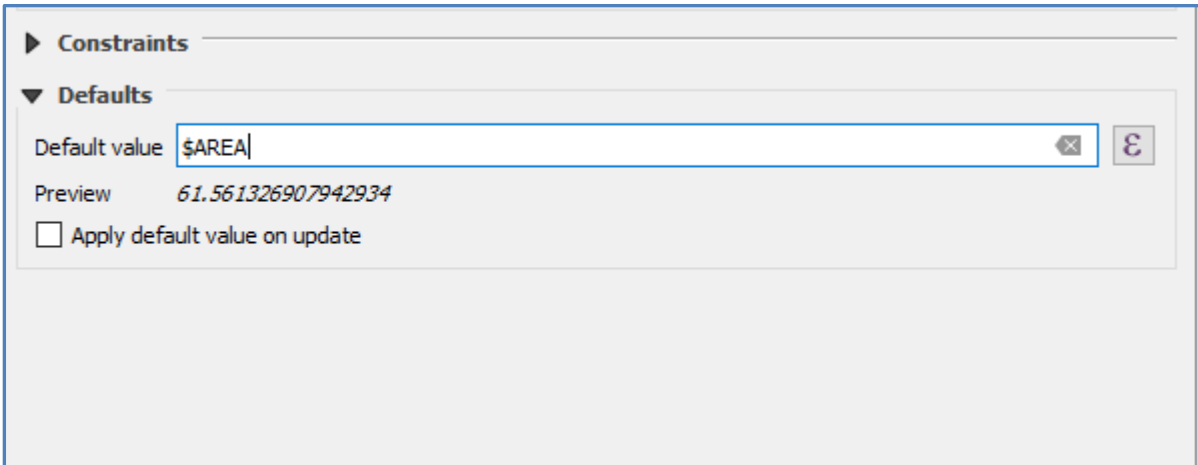
Once within the Attributes Form window, firstly choose the Field that you wish to configure. In this case we will choose the **Area** field.

We want the user to be able to **see** the Area field, but we don't want them to be able **edit** the field when they create a new record. So, in this case we will choose the Widget type to be **Text Edit**, but we will choose to **untick the Editable option**.



However, to ensure that the application inserts the correct Area value when the user creates a new record, we will set the **default value** to be the geometric Area of the new feature. To do this use the Expression:

\$AREA



► Constraints

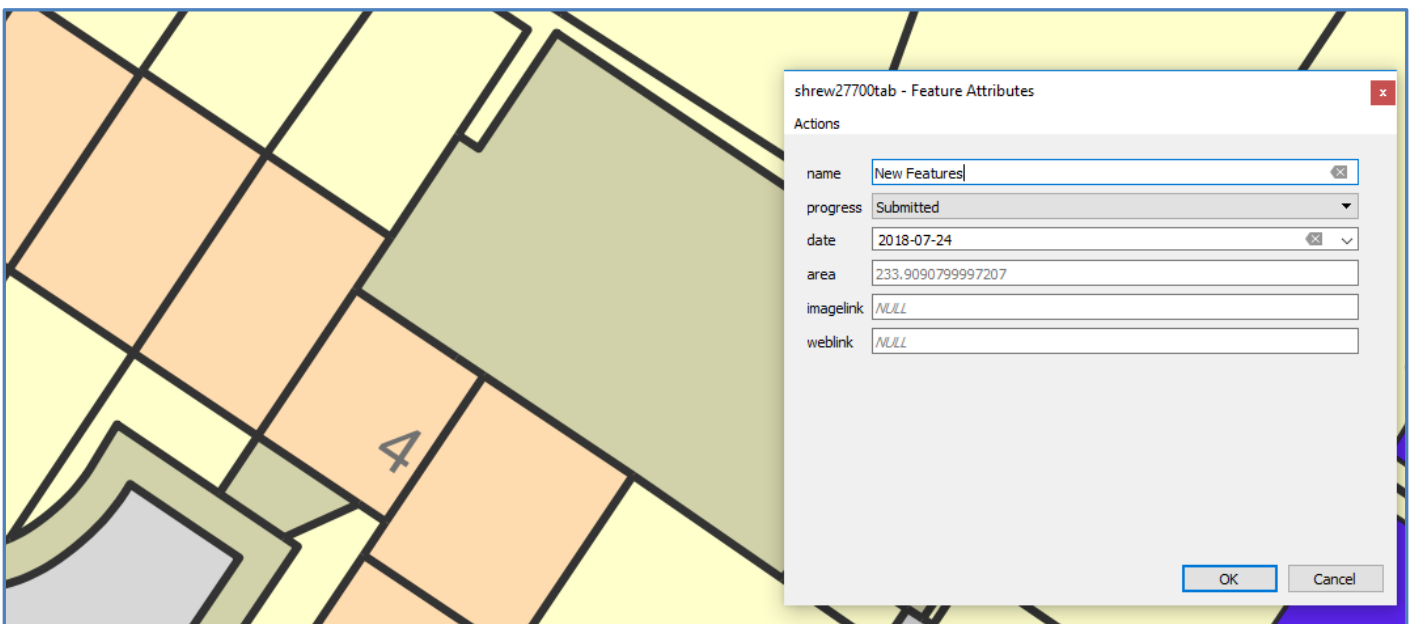
▼ Defaults

Default value

Preview 61.561326907942934

Apply default value on update

Now when the user creates a new feature, the **Edit Attributes form** will show the field **AREA**, but will not allow the user to insert their own value. Instead the system will auto calculate the Area of the new feature and enter that value automatically.

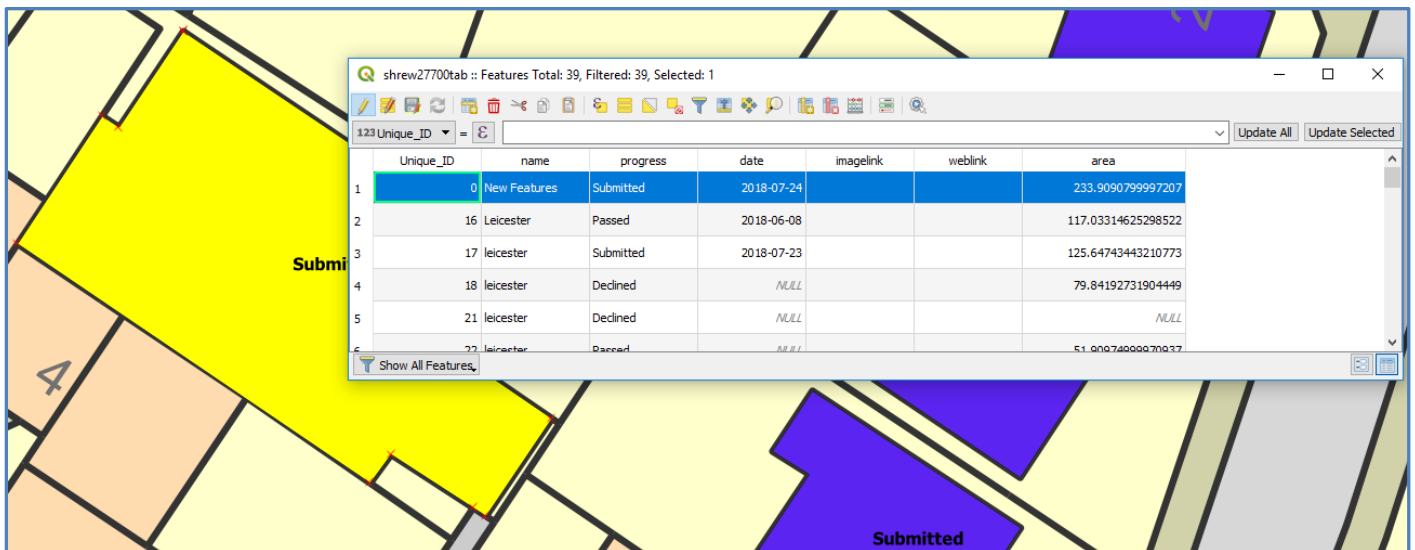


shrew27700tab - Feature Attributes

Actions

name	<input type="text" value="New Features"/>
progress	<input type="text" value="Submitted"/>
date	<input type="text" value="2018-07-24"/>
area	<input type="text" value="233.9090799997207"/>
imagelink	<input type="text" value="NULL"/>
weblink	<input type="text" value="NULL"/>

Once the user completes the record and **saves the layer**, the new feature is added to the layer and the attributes table shows the details of that record, including the newly calculated AREA value.

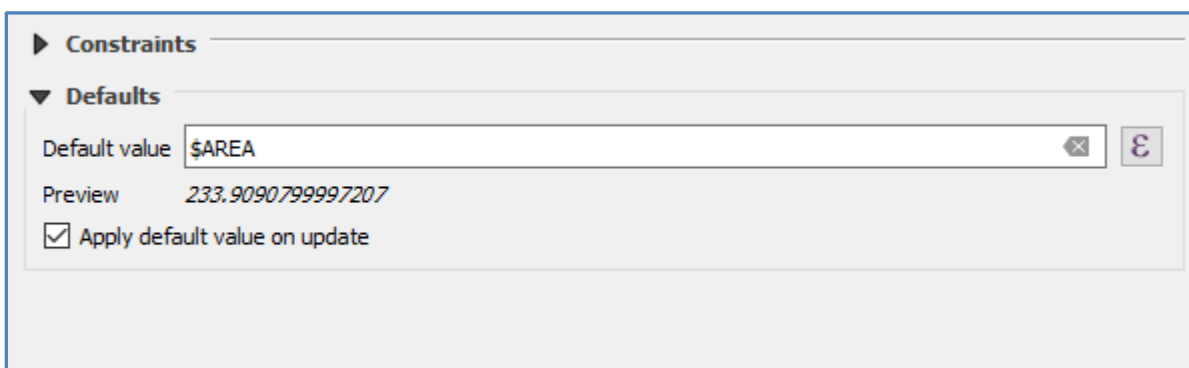


4b – Apply a Default Area Value on Update:

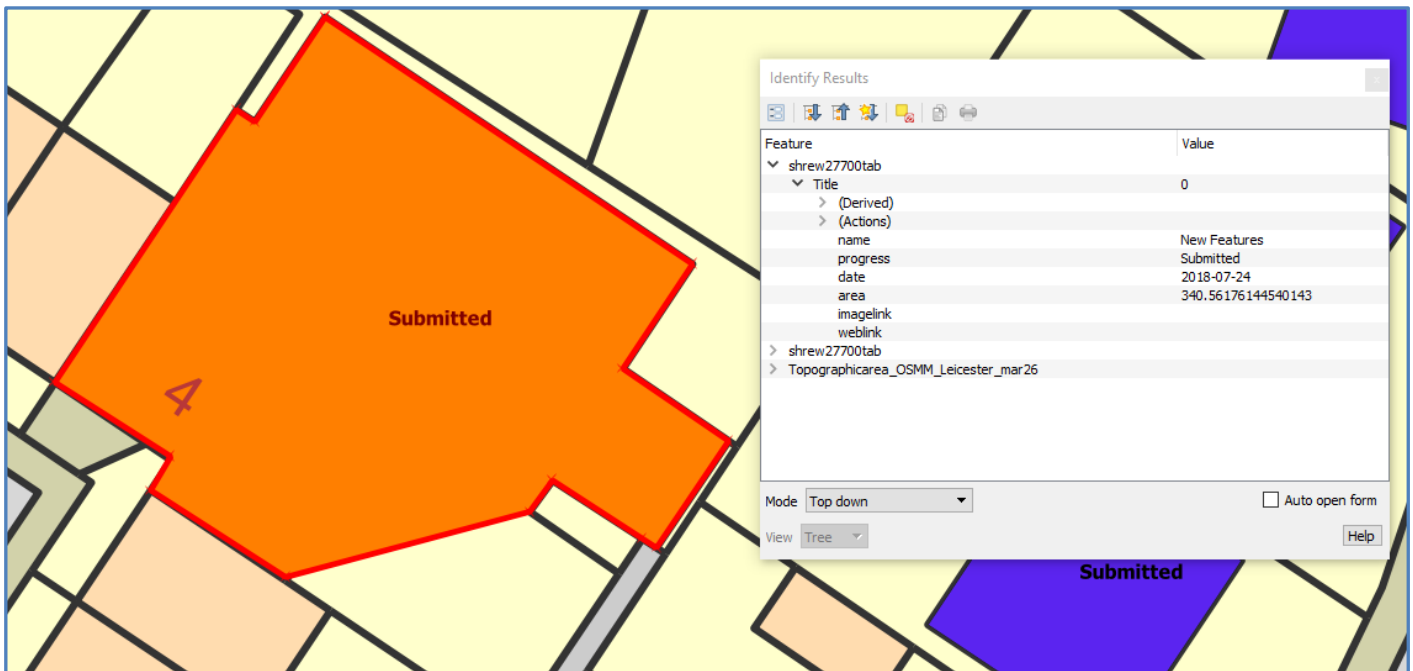
In addition, we can update the **Widget** to automatically update the AREA value whenever a record is **updated**. This can be useful to allow you to ensure that attribute data is kept up to date as the geometry of the spatial records are edited.

Within the **Attributes Form**, firstly select the Field that you wish to automatically create, in this case we will choose the **Area** field.

Having chosen the AREA Field, within the right-hand pane scroll down to the Default options and ensure you tick to **Apply the default value on update** and OK the Layer Properties.



If you now edit an existing feature, e.g. by **reshaping** it, notice that when you save the layer, the Area field will auto update to show the value for the new geometric Area.



Using the Widgets in QGIS, it is very simple to build projects that make data entry and the update of your spatial data as simple and accurate as possible.