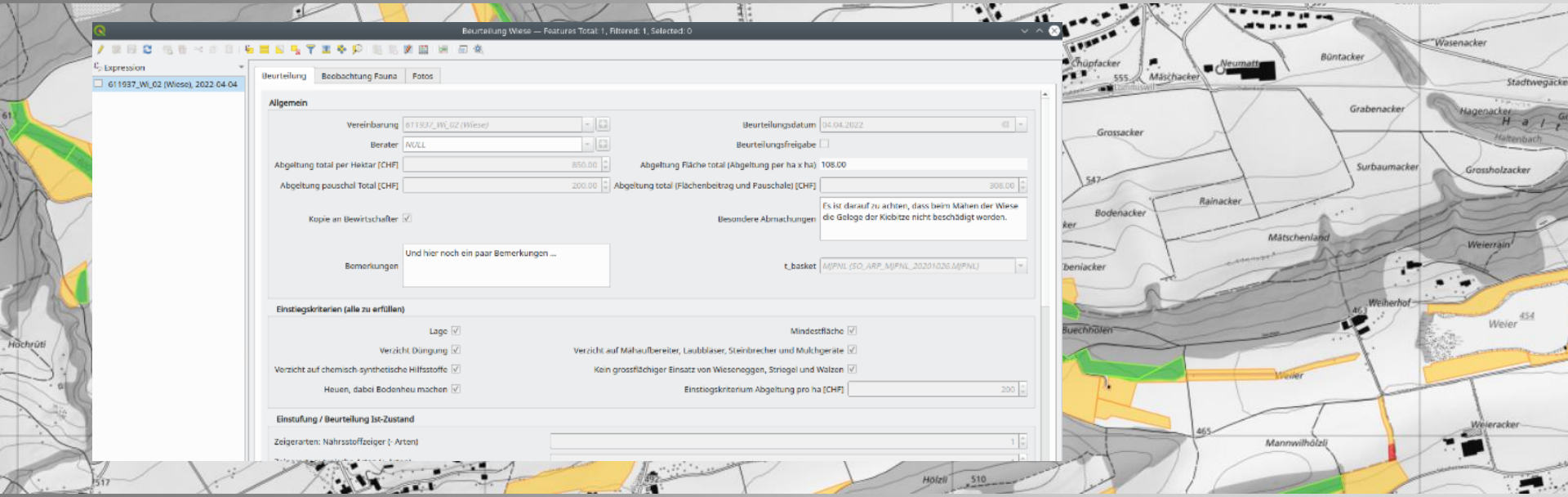
KANTON solothurn

Amt für Geoinformation
Amt für Raumplanung



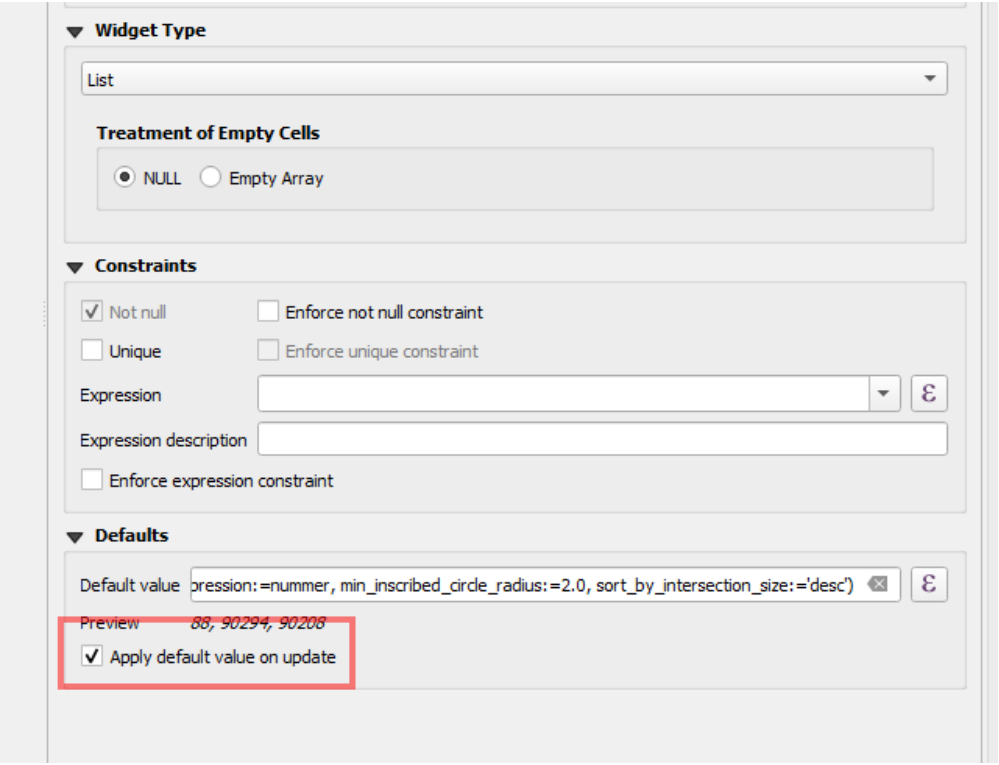
Dynamic Forms and Widgets with QGIS Expressions

By Example of «Mehrjahresprogramm Natur und Landschaft» Kt. Solothurn

Outline

- Default values and updates
- Constraints dependent on other widget values
- Data defined widget visibility
- More sophisticated overlay functions
- Sending form data and map extents to external systems (reporting)

Default values and updates

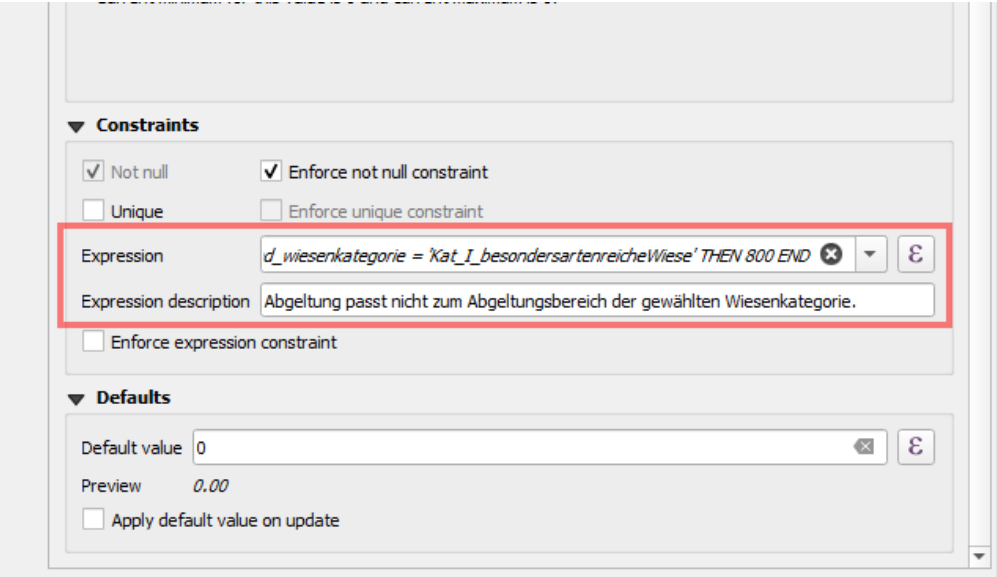


If «Apply default value on update» is enabled, then this widget can react on changes in other fields (wigets)

Multiple dependencies (one dependent widget depending on another dependent widget) should be avoided

Flexible constraints with expressions – dependent on other fields / widgets

KANTON solothurn
Amt für Geoinformation
Amt für Raumplanung



Constraint on «Abgeltung» (compensation) is dependent on «Wiesenkategorie» (grassland category)

Result of expression needs to result in 0 or 1 [true|false]

Limitation: expression description (explaining a constraint violation) cannot be data-defined.

Flexible constraints with expressions – dependent on other fields / widgets

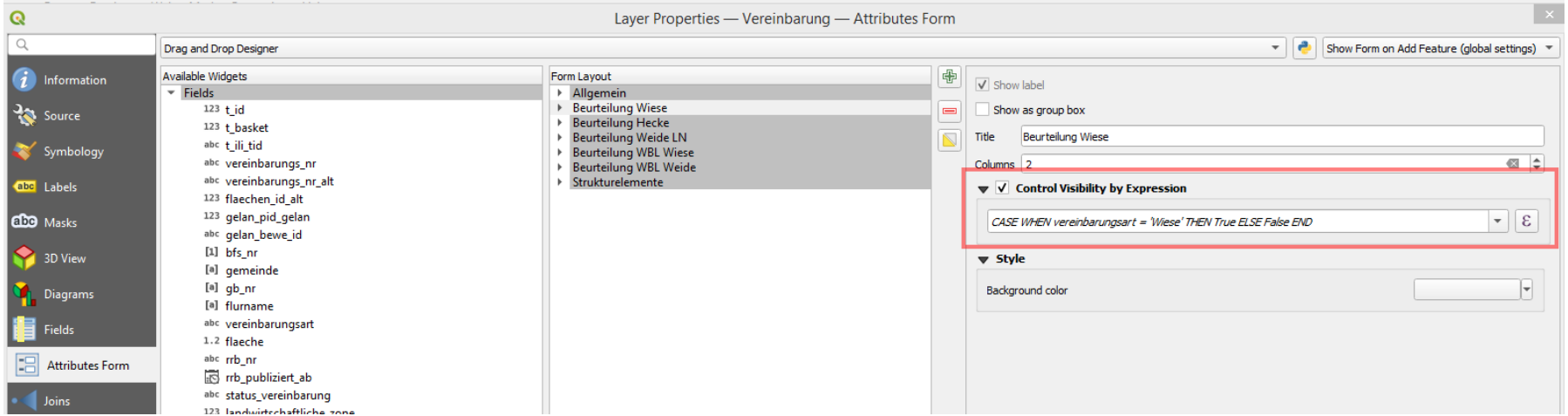

```
wiesenkategorie_abgeltung_ha >=
CASE
   WHEN wiesenkategorie = 'Kat_RF' THEN 0
   WHEN wiesenkategorie = 'Kat_RF_II' THEN 100
   WHEN wiesenkategorie = 'Kat_II_RF' THEN 200
   WHEN wiesenkategorie = 'Kat_II_artenreicheWiese' THEN 400
   WHEN wiesenkategorie = 'Kat_I_besondersartenreicheWiese' THEN 600
END
AND
wiesenkategorie_abgeltung_ha <=
CASE
   WHEN wiesenkategorie = 'Kat_RF' THEN 0
   WHEN wiesenkategorie = 'Kat_RF_II' THEN 100
   WHEN wiesenkategorie = 'Kat_II_RF' THEN 300
   WHEN wiesenkategorie = 'Kat_II_artenreicheWiese' THEN 500
   WHEN wiesenkategorie = 'Kat_I_besondersartenreicheWiese' THEN 800
END
```

lower bound

upper bound

Result of Expression needs to be 0|1 or True|False

Data defined visibility of Group Containers



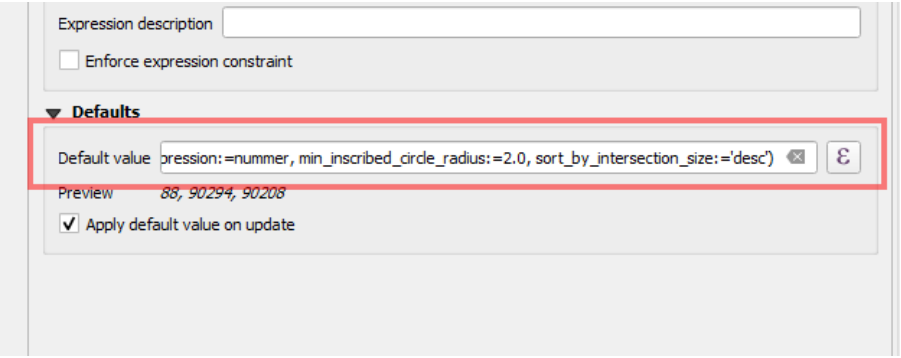
Works for «tabs» and «group boxes» (not for single widgets)

Can be dependent on other fields / widgets

More sophisticated overlay functions

Get proper overlay results (spatial relations).

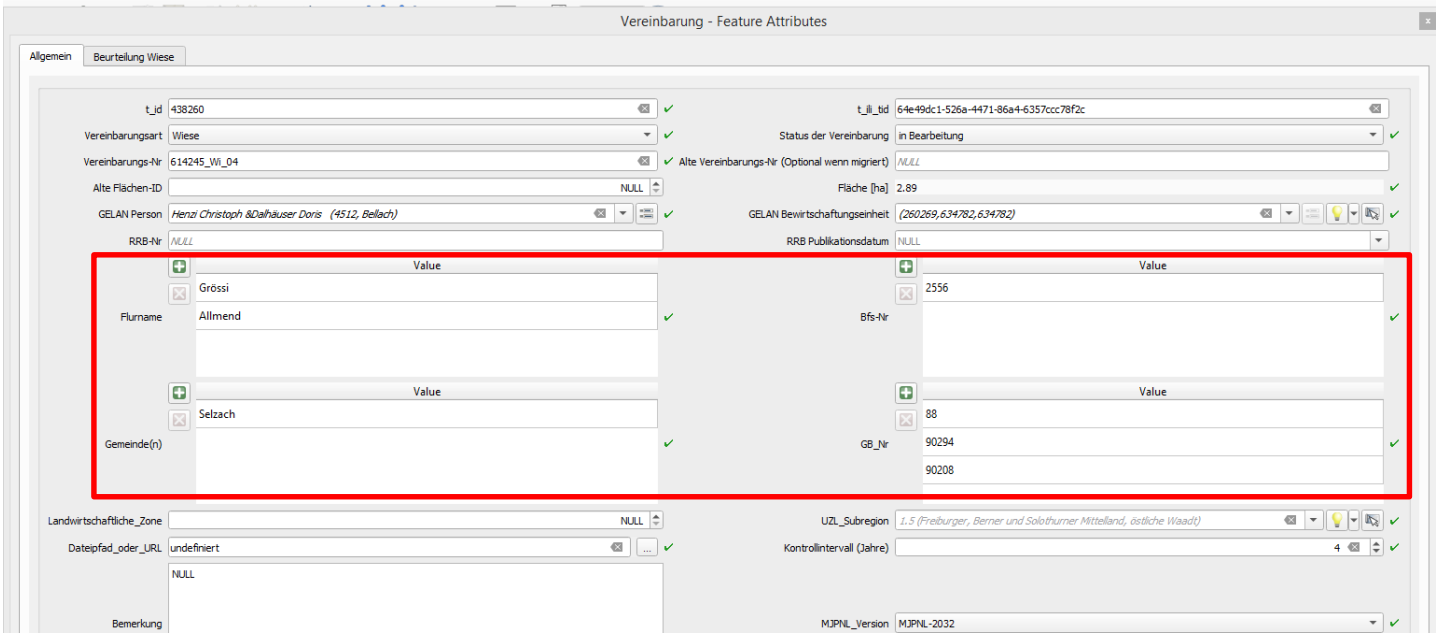
Inclusion criteria:
- min_overlap (area in map units)
- min_inscribed_circle_radius (the maximum possible inscribed circle in the intersection area)



Optional:
- *«sort_by_intersection_size» (descending/ascending)*
- *«return_details»: get back «map» data structure with «feature_id», «expression_result», «overlay_area», «max_inscribed_circle_radius»*

More sophisticated overlay functions

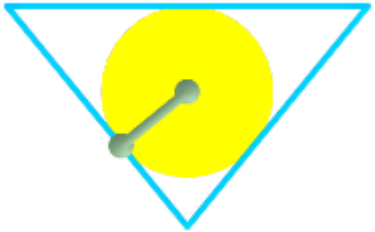
Example: automatically assign municipality, Flurnamen, parcel numbers, etc. through spatial relations



More sophisticated overlay functions

Example: expression to automatically assign Flurnamen based on spatial relations

```
overlay_intersects(
  layer:='AV Flurname',
  expression:=flurname,
  min_inscribed_circle_radius:=3,
  sort_by_intersection_size:='desc'
)
```



Example of minimum inscribed circle in a polygon

More sophisticated overlay functions

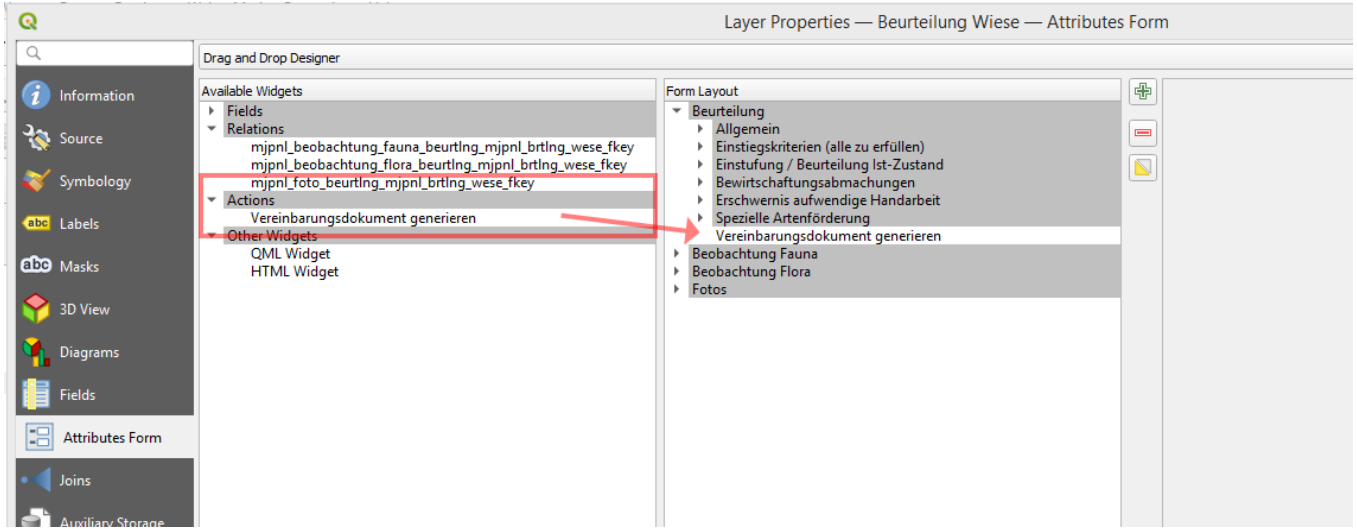

```
attribute(
  get_feature(
    'GELAN Person',
    't_id',
    attribute(
      get_feature(
        'Betrieb',
        't_id',
        attribute(
          get_feature(
            'Bewirtschaftungseinheit',
            'bewe_id',
            array_first(
              overlay_intersects(
                layer:='Bewirtschaftungseinheit',
                expression:="bewe_id",
                sort_by_intersection_size:='des',
                limit:=1
              )
            )
          ),
          'betrieb'
        )
      ),
      'person'
    )
  ),
  'pid_gelan'
)
```

Example: get «Person» (farmer) through overlay function and several relations:

Bewirtschaftungseinheit → Betrieb → Person

Actions as Buttons in Forms

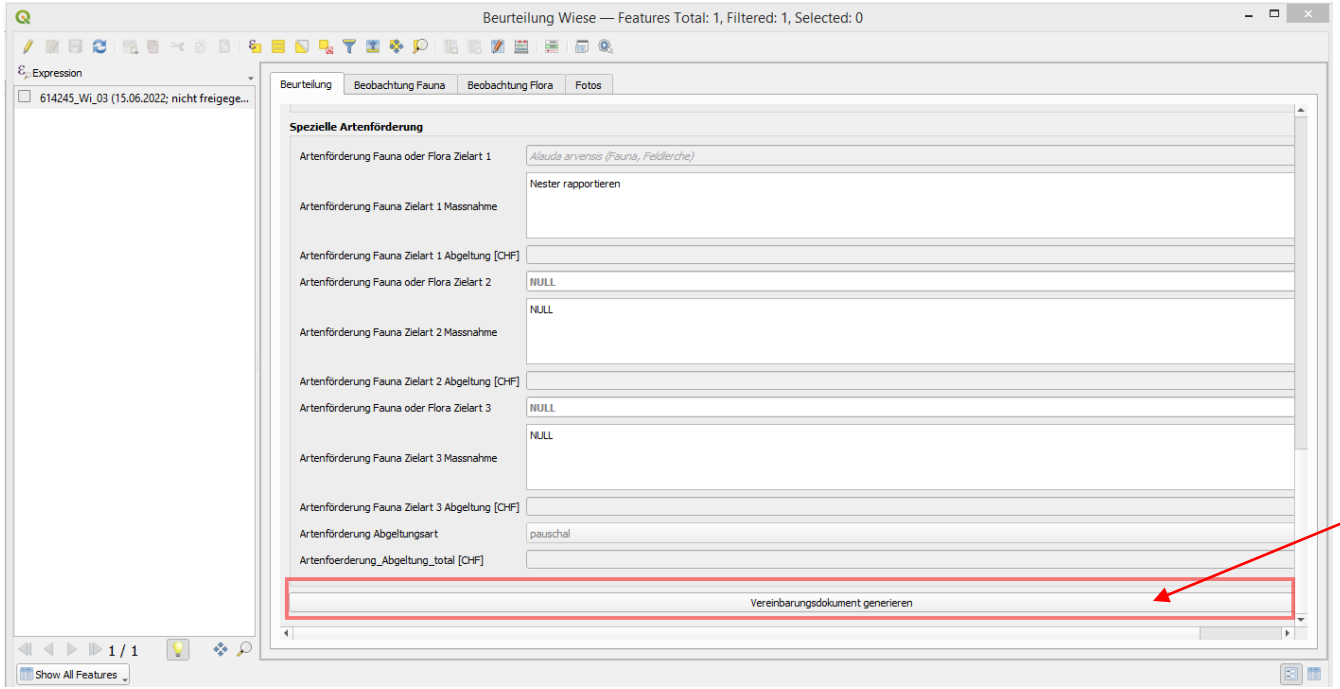
All actions can be assigned to a button and used in drag and drop form



For the definition of the action see the following slides

Actions as Buttons in Forms

All actions can be assigned to a button and used in drag and drop form



This button starts the action

Web-Service calls using actions

Two new action types for POST requests:
- Submit URL (multipart)
- Submit URL (url encoded or JSON)

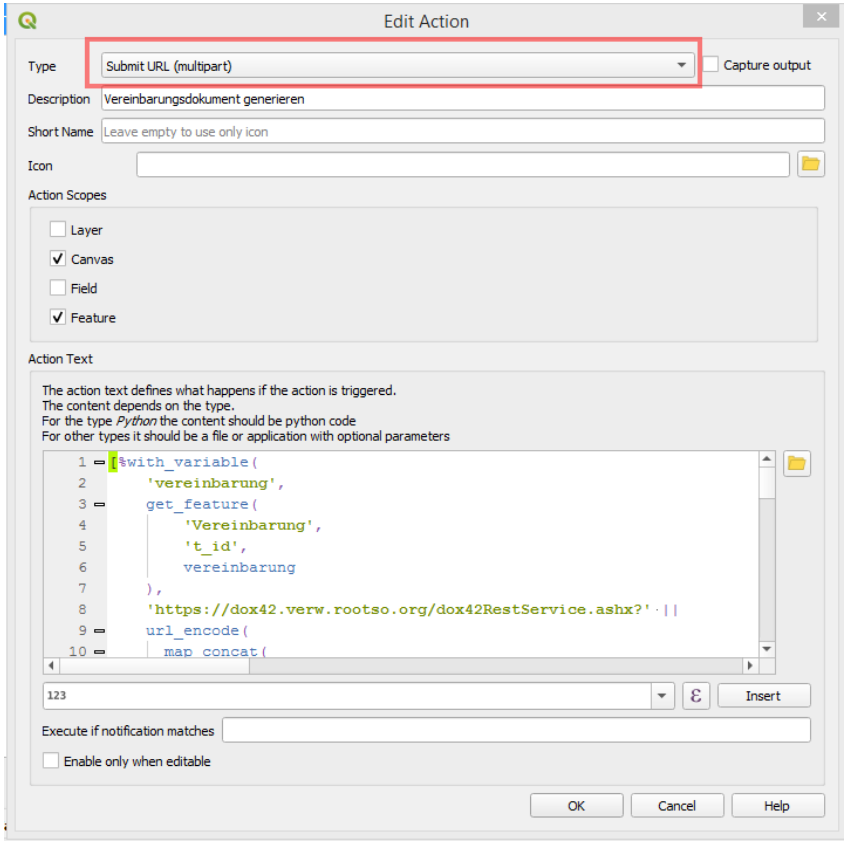
Expression function

`represent_attributes(feature)`

returns all attributes in a «map» data structure (key/values)

`url_encode()`

Encodes non-ascii characters



Atlas images sent to web service

Requires Plugin «AtlasExportFunction» and
a layout with an atlas

New expression function:

```
atlas_image(
        dpi:=150,
        layout_name:='Karte_Detail',
        image_format:= png
)
```

Can be sent as url_encoded binary data through

Code to send attribute data with relations to external Reporting service through Post-URLs

KANTON solothurn

Amt für Geoinformation
Amt für Raumplanung

```
with_variable(
  -- because we need a reference to "vereinbarung"
  -- multiple times we store it in a temporary variable
  'vereinbarung',
  get_feature(
    'Vereinbarung',
    't_id',
    vereinbarung
  ),
  -- base URL of external dox42 reporting service
  'https://dox42.verw.rootso.org/dox42RestService.ashx?' ||
  url_encode(
    map_concat(
      -- Basis URL und Parameter mit Bildern
      map(
        'Operation','GenerateDocument',
        -- template Word file (Path and file name)
        'DocTemplate','c%3A\\dox42Server\\templates\\Wiese.docx',
        'ReturnAction.Format','pdf',
        -- atlas print Detailkarte (default = format png)
        'InputParam.v_karte_detail',to_base64(
          atlas_image(
            dpi:=150,
            layout_name:='Karte_Detail'
          )
        ),
        -- atlas print Übersichtskarte
        'InputParam.v_karte_uebersicht',to_base64(
          atlas_image(
            dpi:=150,
            layout_name:='Karte_Uebersicht'
          )
        )
      ), -- end of first "map" (Basis-URL und Bilder)
      -- Attribute Beurteilung (Wiese)
      map_prefix_keys(represent_attributes($currentfeature),'InputParam.b_'),
      -- Attribute der Vereinbarung
      map_prefix_keys(
        -- gelan_bewe_id auf Originalwert (nicht Repräsentation) zurücksetzen
        map_insert(
          map_delete(
            -- gelan_pid_gelan auf Originalwert (nicht Repräsentation) zurücksetzen
            map_insert(
              map_delete(
                represent_attributes(
                  'Vereinbarung',
                  @vereinbarung
                ),
                'gelan_pid_gelan'
              ),
              'gelan_pid_gelan',
              attribute(
                @vereinbarung,
                'gelan_pid_gelan'
              )
            ),
            'gelan_bewe_id'
          ),
          'gelan_bewe_id',
          attribute(
            @vereinbarung,
            'gelan_bewe_id'
          )
        ),
        'InputParam.v_'
      ),
      -- Attribute der GELAN Person
      map_prefix_keys(
        represent_attributes(
          'GELAN Person',
          get_feature(
            'GELAN Person',
            'pid_gelan',
            attribute(
              @vereinbarung,
              'gelan_pid_gelan'
            )
          )
        ),
        'InputParam.p_'
      )
    ) -- end of map_concat
  ) - end of url_encode
) - end of with_variable
```