



goAML Web

Web Report: Semi-Automation

Created by:

Software Products for Member States
Enterprise Applications Center - Vienna
United Nations

Revision History

Version	Date	Author	Status	Comment
0.1	20-08-2019	Edward Nuttall	Draft	First version. To be added to the goAML Web Manual
0.2	16-09-2019	Thorsten Löhndorf	Draft	Minor fixes.
0.3	07-10-2019	Thorsten Löhndorf	Published	Minor fixes.
0.4	26-11-2019	Thorsten Löhndorf	Published	Added download options for transaction schema
0.5	28-02-2020	Thorsten Löhndorf	In review	Fixed some of the figures, clarified some points, added some additional references

0.6	06-03-2020	Thorsten Löhndorf	Published	Integrated feedback
-----	------------	-------------------	-----------	---------------------

1. Overview

The goAML Web Form provides the ability to upload and download full and partial transactions as XML files. This allows users to create and re-use templates that can speed up the manual entry of reports.

2. How to Download a Transaction

Figure 1 shows a goAMLWeb report that has been partially completed. It contains a single bi-party transaction with an account. It may be required that the user wishes to submit several bi-party transactions that contain this account and will therefore want to download the transaction as it is so that it can be re-used as a template.

Any transactions can be downloaded, it is not necessary for a transaction to be valid or complete.

In order to download the transaction as XML, hover the mouse over the Transaction in the navigator pane on the left of the report and the download icon  will appear. Hovering over this will show the tooltip **Download Transaction (XML)** or the equivalent translation for the selected culture.

Clicking this button will download the specific transaction.

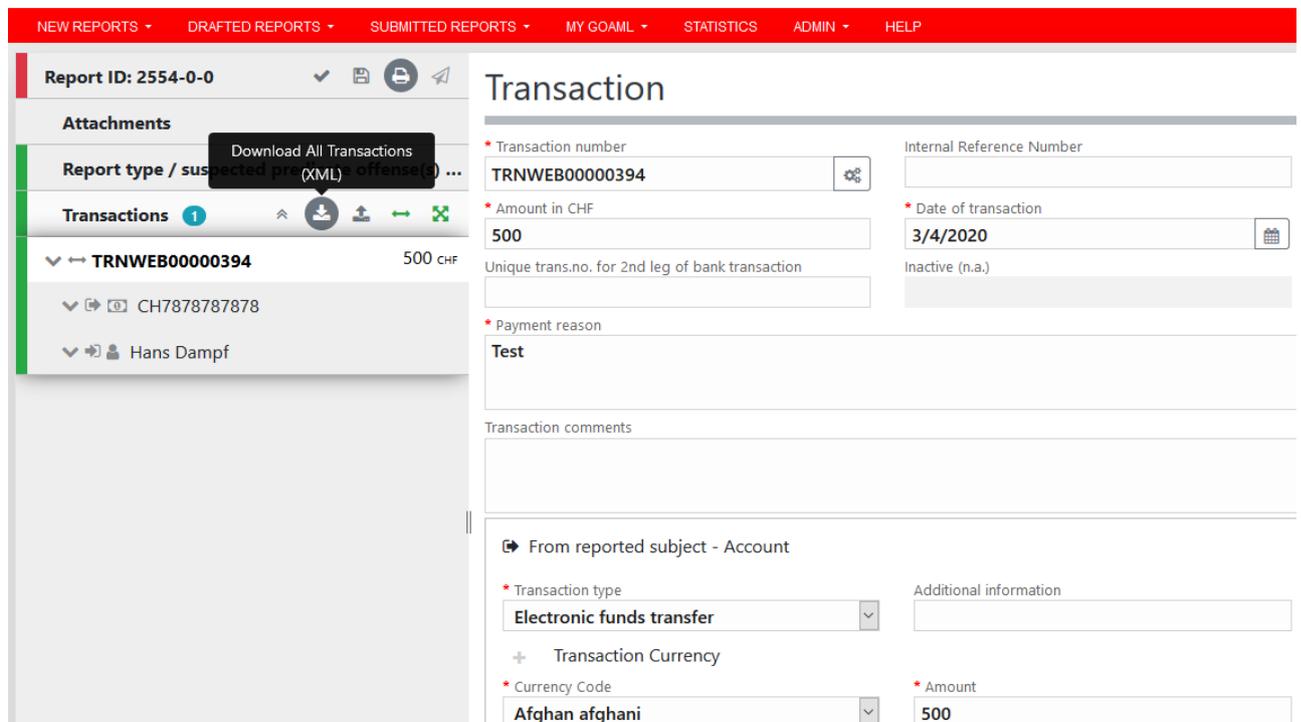


Figure 1 - Transaction ready for download

Figure 2 below shows an example of a resulting XML that is downloaded. Only the fields that are provided in the report are given in the XML. Some (but not necessarily *all*) fields that are mandatory, such as `<transactionnumber>` are also included, but are sometimes provided as an empty tag. It is strongly recommended to fill in at least dummy values for mandatory fields in order to guarantee that they indeed show up in the XML.

```

<?xml version="1.0" encoding="UTF-8"?>
<reportdata>
  <transactions>
    <transaction>
      <transactionnumber>ZV20191231/8091155/1</transactionnumber>
      <transaction_description>Privatkauf</transaction_description>
      <date_transaction>2019-12-31T00:00:00</date_transaction>
      <transmode_code>19</transmode_code>
      <amount_local>550</amount_local>
      <t_from>
        <from_funds_code>4</from_funds_code>
        <from_foreign_currency>
          <foreign_currency_code>CHF</foreign_currency_code>
          <foreign_amount>550</foreign_amount>
          <foreign_exchange_rate>1</foreign_exchange_rate>
        </from_foreign_currency>
        <from_account>
          <institution_name>Zürcher Stadtbank</institution_name>
          <swift>ZSTCHZ80A</swift>
          <account>CH2000</account>
          <signatory>
            <is_primary>true</is_primary>
            <t_person>
              <gender>M</gender>
              <first_name>Fritz</first_name>
              <last_name>Caspar</last_name>
              <addresses>
                <address>
                  <address_type>1</address_type>
                  <address>Kries</address>
                  <city>Dueb</city>
                  <zip>8600</zip>
                  <country_code>CH</country_code>
                  <state>ZH</state>
                </address>
              </addresses>
            </t_person>
            <role>14</role>
          </signatory>
          <comments>Privatkauf</comments>
        </from_account>
      </t_from>
      <t_to_my_client>
        <to_funds_code>4</to_funds_code>
        <to_foreign_currency>
          <foreign_currency_code>CHF</foreign_currency_code>
          <foreign_amount>550</foreign_amount>
          <foreign_exchange_rate>1</foreign_exchange_rate>
        </to_foreign_currency>
        <to_account>
          <institution_name>Magaroni Bank</institution_name>
          <swift>MAGXXX</swift>
          <branch>AG</branch>
          <account>CH999999999</account>
          <currency_code>CHF</currency_code>
          <iban>CH999999999</iban>
          <client_number>142418</client_number>
          <personal_account_type>2</personal_account_type>
          <signatory>
            <is_primary>true</is_primary>
            <t_person>
              <gender>W</gender>
              <first_name>Frane</first_name>
              <last_name>Joggi</last_name>
              <birthdate>1958-06-19T00:00:00</birthdate>
              <nationality1>CH</nationality1>
              <addresses>
                <address>
                  <address_type>3</address_type>
                  <address>Alp</address>
                  <city>Kreugen</city>
                  <zip>6200</zip>
                  <country_code>CH</country_code>
                  <state>TG</state>
                </address>
              </addresses>
              <identification>
                <type>2</type>
                <number>E19964</number>
                <issue_date>2014-10-03T00:00:00</issue_date>
                <expiry_date>2024-10-02T00:00:00</expiry_date>
                <issued_by>Herrenfeld TG</issued_by>
                <issue_country>CH</issue_country>
              </identification>
            </t_person>
            <role>19</role>
          </signatory>
          <opened>2019-06-18T00:00:00</opened>
          <closed>2020-01-07T00:00:00</closed>
          <status_code>2</status_code>
          <beneficiary_comment>Kreingen</beneficiary_comment>
        </to_account>
      </t_to_my_client>
    </transaction>
  </transactions>
</reportdata>

```

Figure 2 - Downloaded Transaction XML

Everything inside the `<transaction>` tags follow the structure of the goAML Schema.

The difference between the XML used to download/upload transactions and the complete XML schema is that the transactions are *wrapped* in the `<reportdata>` and `<transactions>` tags. This is because multiple transactions can be downloaded or uploaded at once, and so a root element is required (**reportdata**). The group element (**transactions**) is used so that the functionality can be extensible in the future to accommodate other groups of reusable objects such as Persons, Accounts and Entities.

The schema for the validation of transaction uploads can either be downloaded from the MROS [web page](#) or directly from within goAML. Please note that this XML Report Validator function is *only available* in the **pilot environment** and *not* in the production environment. First navigate to the XML Report Validator (see Figure 3).

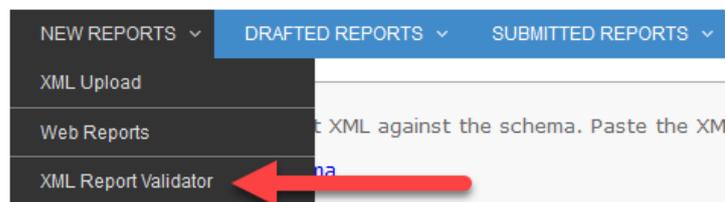


Figure 3: XML Report Validator

Then click on the download link (see Figure 4).

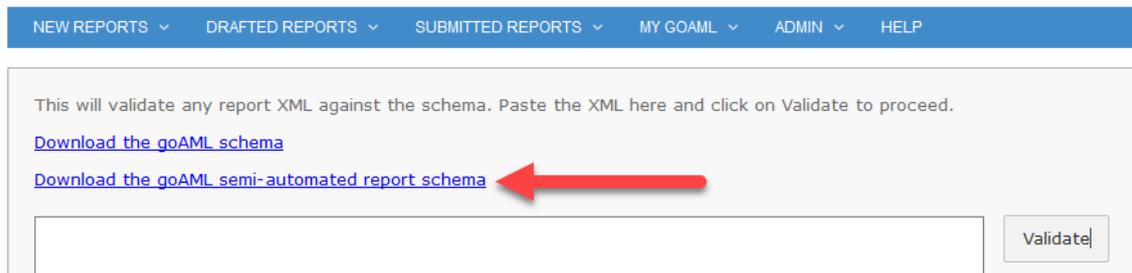


Figure 4: Download link for transaction schema

If you try to implement a system that generates these XML files, it is *important* to ensure that your XML files successfully validate against this XML schema. Otherwise your transaction uploads will fail.

The following XML example shows the basic structure on how multiple transactions will appear when downloaded.

```

<reportdata>
  <transactions>
    <transaction>
      ...
    </transaction>
    ...
    </transaction>
    ...
  </transactions>
</reportdata>
    
```

All transaction files that are downloaded can be used immediately for uploading as templates for new transactions.

Figure 5 shows where to download *all* of the transactions in the report. Hovering the mouse pointer over the **Transactions** header in the left navigator will display the button (see Figure 5). Please note that this button will only be displayed if you have entered at least one transaction.

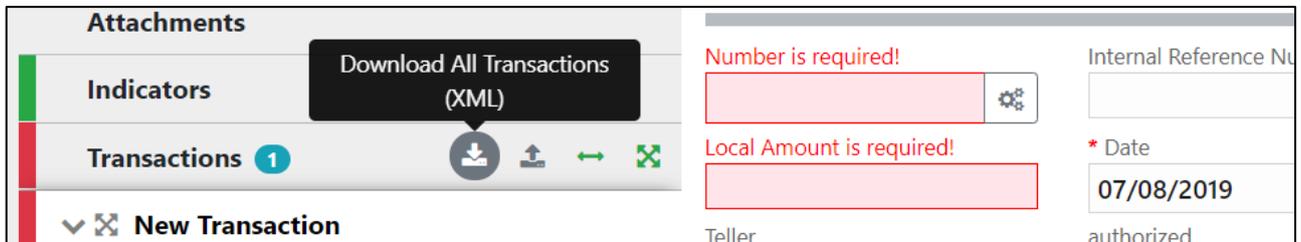


Figure 5 - Download All Transactions

3. How to Upload Transactions

Each transaction should be inside a `<transaction>` tag and must follow the goAML schema structure, however it does not have to be valid, i.e. the data does not have to be complete or conform to the restrictions of the schema such as mandatory, min/max, decimal etc. Nevertheless, it is recommended that it is as compliant with the XML schema as possible. The best and safest way to do this is to validate it against the official XML schema. As stated before it is possible and allowed to upload incomplete transactions but this would necessitate entering the missing values manually after the transaction upload.¹

In the XML file, the transactions should then be placed inside `<reportdata>` and `<transactions>` tags as seen in Figure 6. It is also possible to upload each transaction individually in separate files. However even a single transaction must be contained in these *wrapper* tags for it to be uploaded successfully.

```

<reportdata>
  <transactions>
    <transaction>
      <transactionnumber>ZV20191231/0091155/1</transactionnumber>
    </transaction>
  </transactions>
</reportdata>
    
```

↕

```

    </transaction>
  </transactions>
</reportdata>
    
```

¹ Please note that being validated against the XML schema is the *first* step of getting a report accepted. This validation checks that the form and structure of the XML are acceptable. The content will be checked against the business rules after report submission. The business rules are described in the document "[Standard XML Reporting - Instructions and Specifications for goAML](#)".

Figure 6 - Upload All Transactions

Once the file is ready for upload, place the mouse over the **Transactions** header in the left hand navigator so that the **Upload Transactions (XML)**- button is displayed as shown in Figure 7 and click it to open the file dialog.

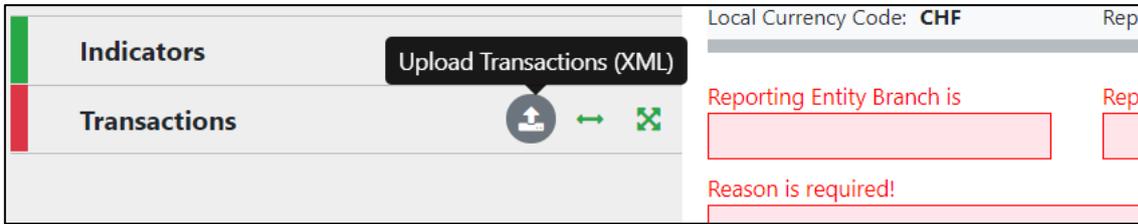


Figure 7 - Upload Transactions button

Select the file containing the transactions and click the **Upload** button as shown in Figure 8.

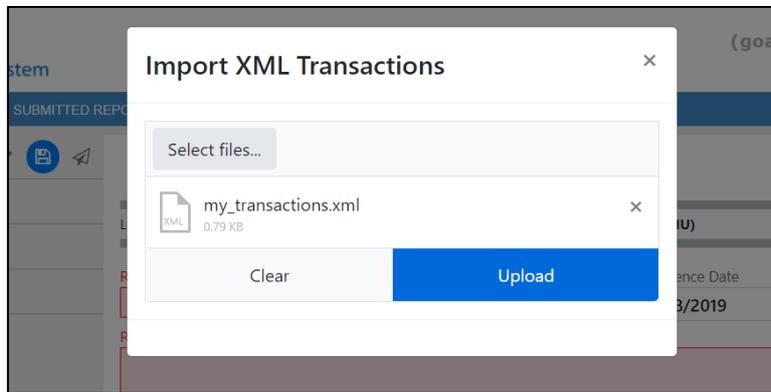


Figure 8 - Upload XML Transactions Dialog

The content of the file will be read and the transactions will be added to the report as shown in Figure 9.

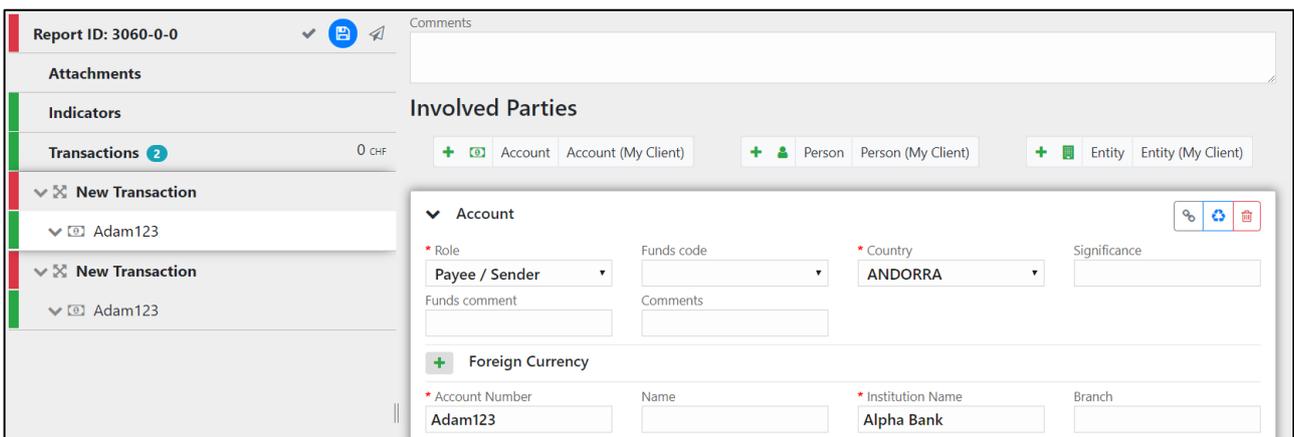


Figure 9 - Report after transactions upload

All transactions are added to the report as they are shown in the file. There is no processing on the report or file. For example, if there is an existing transaction in the report with the same number as one in the file, it will result in two transactions in the report with the same number. Any linking of accounts must also be done manually after the transactions have been uploaded.

4. Linking of Accounts

After transactions have been uploaded, initially all objects such as Persons, Accounts and Entities are treated individually, that means that changing the data on one object will not change the data on any of the other objects even if they have exactly the same data.

For Accounts, it is possible to link objects that share the same **Account Number** so that any changes to one of those Account objects will be reflected in all the others.

For example:

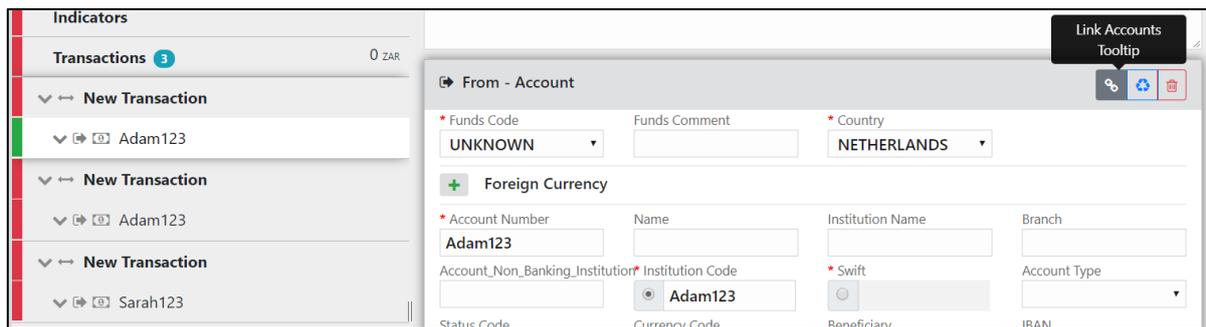


Figure 10 - Linking Accounts

In Figure 10 there are three uploaded transactions each with an account. The first two accounts have the same account number **Adam123**. The first account is selected and highlighted in the editor.

By clicking on the **Link Accounts** button  of the first account, all of the information of the first account will be copied to the second account (and any other accounts that have this number). After this, any change to either of these accounts will be reflected in the other. The third account of **Sarah123** is not affected.

The green box in Figure 11 shows the areas that contain the linked account data. The red boxes show **From** and **Foreign Currency** objects which are not part of the **Account** object and are therefore not linked.

From - Account

🔗
🔄
🗑️

* Funds Code UNKNOWN	Funds Comment	* Country NETHERLANDS
--------------------------------	---------------	---------------------------------

+ Foreign Currency

* Account Number Adam123	Name	Institution Name	Branch
Account_Non_Banking_Institution	* Institution Code Adam1236	* Swift	Account Type
Status Code	Currency Code	Beneficiary	IBAN
Client Number	Opened M/d/yyyy	Closed M/d/yyyy	Balance
Date of Balance M/d/yyyy	Beneficiary Comment	Comments	

+ Entity

+ Signatory(ies)

Figure 11 - Account data