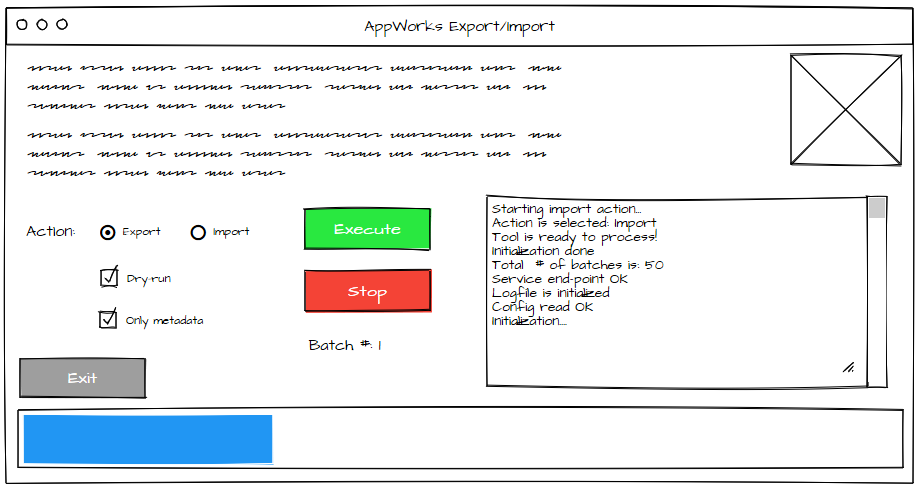
AppWorks Export/Import tool

Use-cases

# Package definitions

1. **General**
   1. Start the tool
   2. Do validation
   3. Start export
   4. Stop export
   5. Start import
   6. Stop import
   7. Switch for a dry-run
   8. Switch for ‘only metadata’
   9. Monitor current workload
   10. Process the selected action
   11. Check the current/next item
2. **Storage**
   1. Write service
   2. Read service
   3. Write content
   4. Read content
   5. Write metadata
   6. Read metadata
   7. Read configuration
   8. Validate mapping file
3. **Messaging**
   1. Show notification
   2. Write to log file
   3. Send a mail
   4. Read mail template

# UI wireframe example



# Generic package

### Start the tool

#### Sunny-day

The user starts the tool, show the main screen and the tool will do the first initialization

#### Rainy-day

One of the initialization checks fail

### Do validation

#### Sunny-day

After startup the tool will do the first initialization with the following checks

* Configuration is valid and done correctly
* Service end-point is reachable with the correct credentials
* XML mapping file is valid
* Logfiles are ready
* Total # of batches is detected
* Storage is reachable for read/write
* Mail server available
* Mail template available

#### Rainy-day

* The configuration can’t be found
* The configuration is invalid
* Service end-point is invalid
* One of the initialization checks fail

### Start export

#### Sunny-day

The user selects the action to be executed and clicks on the execute button. The tool starts to process the export action

#### Rainy-day

* The action to be selected is not available
* The execute button is not available
* The initialization of the tool is OK, but before executing the service end-point it unreachable

### Stop export

#### Sunny-day

The user clicks the stop button. The tool finishes the current workload and stops further processing

#### Rainy-day

* The user did not start the export yet
* The tool is busy with a current workload

### Start import

#### Sunny-day

The user selects the action to be executed and clicks on the execute button. The tool starts to process the import action

#### Rainy-day

* The action to be selected is not available
* The execute button is not available
* The initialization of the tool is OK, but before executing the service end-point it unreachable

### Stop import

#### Sunny-day

The user clicks the stop button. The tool finishes the current workload and stops further processing

#### Rainy-day

* The user did not start the import yet
* The tool is busy with a current workload

### Switch for a dry-run

#### Sunny-day

The user marks the dry-run option. The tool will adapt the execution based on this setting so only logging is written away.

#### Rainy-day

* Log file is not available

### Switch for ‘only metadata’

#### Sunny-day

The user marks the only metadata option. The tool will adapt the execution based on this setting so only metadata written away.

#### Rainy-day

* Filestorage is not reachable

### Monitor current workload

#### Sunny-day

When the user starts an action, the tool starts executing the configured workload (= the current batch size). The current batch number is shown to the user and the progress bar is updated

#### Rainy-day

* Logfiles are unavailable

### Process the selected action

#### Sunny-day

After the user starts the selected action to tool will process the selected action.

#### Rainy-day

* Service end-points is unreachable

### Check the current/next item

#### Sunny-day

When the tool is processing the current workload the current and next items are available for viewing and checking the status.

#### Rainy-day

* Current status is unavailable
* Next item can’t be found
* Last item is reached

# Storage

### Write service

#### Sunny-day

The tool will start writing/importing data (metadata and/or content) to the service end-point when the batches are processed. The input data will be read from the storage location

#### Rainy-day

* The storage is unreachable
* The service is unavailable
* The dry-run option is enabled

### Read service

#### Sunny-day

The tool will start reading/exporting data (metadata and/or content) from the service end-point when the batches are processed. The output data will be written to the storage location

#### Rainy-day

* The storage is unreachable
* The storage is read-only
* No access to the storage
* The service is unavailable

### Write content

#### Sunny-day

The tool will start writing content to the storage when the batches are processed. The storage will be an export location and the data is retrieved (exported) from the read activity from the service

#### Rainy-day

* The storage is unreachable
* The storage is read only
* The dry-run option is enabled

### Read content

#### Sunny-day

The tool will start reading content from the storage when the batches are processed. The storage will be an import location and the data is uploaded to the write activity of the service

#### Rainy-day

* The storage is unreachable
* No access to the storage

### Write XML metadata

#### Sunny-day

The tool will start writing metadata to the storage (as XML) when the batches are processed. The storage will be an export location and the data is retrieved (exported) from the read activity from the service

#### Rainy-day

* XML files not available
* No access to the files
* The dry-run option is enabled

### Read XML metadata

#### Sunny-day

The tool will start reading (XML) metadata from the storage when the batches are processed. The storage will be an import location and the data is uploaded to the write activity of the service

#### Rainy-day

* XML files not available
* No access to the files
* XML data is corrupted
* XML data is not correctly mapped to the service end-point

### Read configuration

#### Sunny-day

On tool initialization the configuration file will be read so the tool knows how to interact. These settings should be readable options

* Service end-point (connection string)
* Batch size
* Nr. Of threads
* Storage location
* Retry value
* Log level
* Mapping XML
* Mail server
* Mail credentials
* Mail template

#### Rainy-day

* Configuration file is not available
* Configuration is invalid

### Validate mapping file

#### Sunny-day

The tool will use a mapping file that makes sure the external data can be correctly mapped to the internal data. A sample layout for the mapping can be found in the end of this document

#### Rainy-day

* Mapping file not available
* Mapping file not in correct format
* Mapping file not correctly mapped

# Messaging

### Show notification

#### Sunny-day

When the tool gets an error, the user will be informed via a message in the information box/pop-up

### Write to log file

#### Sunny-day

Per processed item a log entry will be written to a log file

#### Rainy-day

* Log file not found

### Send a mail

#### Sunny-day

When the tool is done with the executed action a mail message will be send out to the configured mail configuration

#### Rainy-day

* Mail server unreachable
* Mail credentials invalid

### Read mail template

#### Sunny-day

When the tool sends a mail, a configured template is used to send out the information

#### Rainy-day

* Mail template not available

# Property file example

service.url = http://hostname:8080/home/system/services

service.user\_name = sysadmin

service.password = admin

batch\_size = 10

storage = //share01/c$/export

retries\_on\_failure = 3

log\_level = INFO

mapping\_xml = //share01/c$/config/mapping.xml

mail.server = mail.company.com

mail.user\_name = admin

mail.password = admin

mail.template = //share01/c$/config/template.msg

scaling\_size = 1

# <mapping xmlns:xsi="http://www.w3.org/2001/XMLScXML mapping example

<?xml version="1.0" encoding="UTF-8" ?>

<mapping xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="mapping.xsd">

<entity>

<name>category</name>

<can\_have\_content>true</can\_have\_content>

<properties>

<property>

<from>cat\_name</from>

<to>cat\_name</to>

</property>

<property>

<from>cat\_description</from>

<to>cat\_description</to>

</property>

</properties>

</entity>

<entity>

<name>order</name>

<can\_have\_content>false</can\_have\_content>

<properties>

<property>

<from>order\_name</from>

<to>order\_name</to>

</property>

<property>

<from>order\_quantity</from>

<to>order\_quantity</to>

</property>

</properties>

</entity>

</mapping>

## Corresponding XSD

<https://www.liquid-technologies.com/online-xml-to-xsd-converter>

<?xml version="1.0" encoding="UTF-8" ?>

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

<xs:element name="mapping">

<xs:complexType>

<xs:sequence>

<xs:element maxOccurs="unbounded" name="entity">

<xs:complexType>

<xs:sequence>

<xs:element name="name" type="xs:string" />

<xs:element name="can\_have\_content" type="xs:boolean" />

<xs:element name="properties">

<xs:complexType>

<xs:sequence>

<xs:element maxOccurs="unbounded" name="property">

<xs:complexType>

<xs:sequence>

<xs:element name="from" type="xs:string" />

<xs:element name="to" type="xs:string" />

</xs:sequence>

</xs:complexType>

</xs:element>

</xs:sequence>

</xs:complexType>

</xs:element>

</xs:sequence>

</xs:complexType>

</xs:element>

</xs:sequence>

</xs:complexType>

</xs:element>

</xs:schema>