

# 3D-SYSTEMEK Reposit

## User Manual

Version 3.4.0



### Authors:

Christos Georgis ([georgis@ics.forth.gr](mailto:georgis@ics.forth.gr))

Ioannis Chrysakis ([hrysakis@ics.forth.gr](mailto:hrysakis@ics.forth.gr))

Foundation for Research and Technology, Hellas (FORTH)



ΕΥΡΩΠΑΪΚΗ ΕΝΩΣΗ  
ΕΥΡΩΠΑΪΚΟ ΤΑΜΕΙΟ  
ΠΕΡΙΦΕΡΕΙΑΚΗΣ ΑΝΑΠΤΥΞΗΣ



η περιφέρεια στο επίκεντρο της ανάπτυξης



Υπουργείο Παιδείας και Θρησκευμάτων, Πολιτισμού και Αθλητισμού  
ΓΓΕΤ - ΕΥΔΕ-ΕΤΑΚ

Ε. Π. Ανταγωνιστικότητα και Επιχειρηματικότητα (ΕΠΑΝ ΙΙ), ΠΕΠ Μακεδονίας – Θράκης, ΠΕΠ Κρήτης και Νήσων Αιγαίου, ΠΕΠ Θεσσαλίας – Στερεάς Ελλάδας – Ηπείρου, ΠΕΠ Αττικής



## Table of Contents

<b>1</b>	<b>Introduction .....</b>	<b>5</b>
1.1	Functional Point Characteristics .....	5
1.2	Motivation .....	6
<b>2</b>	<b>Workflow Considerations .....</b>	<b>7</b>
2.1	Describing the output of an Event.....	7
2.2	Describing details of an Event .....	7
2.3	Describing the input of a Process Event .....	7
2.3.1	All input files should be already in the RI .....	7
2.3.2	Introduction of intermediate files .....	7
2.3.3	Some process events may not have input files.....	9
2.4	Describing the documentation of an Event.....	9
2.5	Asynchronous and synchronous ingest.....	9
<b>3</b>	<b>User Interface Description .....</b>	<b>10</b>
3.1	Login Form .....	10
3.2	Handling an Event description.....	11
3.3	Acquisition event .....	12
3.3.1	Output Digital Object .....	12
3.3.2	Acquisition Event .....	14
3.3.3	Setup & Devices .....	16
3.3.4	Documentation .....	17
3.4	Process Event.....	18
3.4.1	Output Digital Object .....	18
3.4.2	Process Event.....	19
3.4.3	Input Digital Objects .....	20
3.4.4	Documentation .....	22
3.5	Forms for the creation and update of entities .....	23
3.6	Form Input fields functionality .....	26
3.6.1	Input field functionality .....	26
3.6.2	Multi-valued Input fields.....	26
3.6.3	Creation and update of entities .....	26
3.6.4	Update of Events .....	27
3.6.5	Creation of new Types .....	27
<b>4</b>	<b>Appendix.....</b>	<b>28</b>
4.1	Tool.properties file .....	28
4.2	User Control Messages.....	28



# 1 Introduction

This document describes a tool for ingesting into the 3D Repository Infrastructure data objects for or from 3D Model production that have not been produced together with valid 3D Digital provenance metadata.

The tool guides the user during the ingest process to manually provide sufficient and consistent information which is captured in metadata. Data objects in 3D-Model production are typically created in individual processes which form part of an open, only partially constraint workflow with complex structure. In order to support the envisaged reasoning functions for data management and scientific interpretation, it is critical for the consistency of the metadata that the output of one process is referred as input in the next process with exactly the same identifier, as well as all persons, places and other things involved in more than one process. The tool however limits the possible complexity of a particular acquisition process to at most one level of sub-processes.

The main challenge of the tool is to ensure this *referential integrity* by querying existing reference data in the repository and validating user input against related data in the repository.

The tool aims further at providing a good-practice library of software components for metadata creation. The tool further allows referring to and loading complete, existing metadata records describing the digitized cultural object.

## 1.1 Functional Point Characteristics

The tool is based on the specifications that arose from Deliverable 1.1. We present the basic supported *event-based* functionality:

- Support of the **ingestion of data of a simple acquisition process**. It provides an automatic way to create the metadata files of an acquisition process, providing the user the appropriate form(s) to assign metadata information on the acquisition event, identification information of the depicted object, and general acquisition setup and capturing device(s) information.
- Support of the **ingestion of data of a simple data-processing procedure**. It provides an automatic way to create the metadata files of a data-processing procedure, providing the user the appropriate form(s) to assign metadata information on the processing event. Involved partners should identify the required metadata for describing S/W and S/W-versions, along with the parameters that are passed to the specific S/W. There is no limitation to the number of independent acquisition processes and subsequent processing steps ingested individually, but contributing to one or more 3D Models.
- **Mechanisms** that assist in the ingestion of metadata information of intermediate-processing-steps that users decide to store in the repository (hereafter RI), while their intermediate data-files are lost or erased.
- **Unified URI policy definition**.
- **Full referential integrity control mechanism**. All forms interact with the RI and get information from it, which is selectable by the user. Thus there are no misspellings, nor illegal cross referencing.
- All forms may be saved and re-loaded for further editing. This enables the user to **revise** the **metadata** before committing in the RI by using **event templates**.
- **Generation of zip files** before upload for convenience usage and better file usage.
- **Asynchronous or synchronous ingest/upload** of all data and metadata in the RI for gaining better performance.
- Auto **re-upload feature** for incomplete files.
- **Information and Resuming** of pending files.

- **Type ahead search** at all fields for easy access to the ingested entities/digital objects of the repository.
- **Rollback mechanisms** for corrupted event or files and auto-switching to update mode if part-of event has been ingested before a failure happens (i.e. network failure).
- **Logging of SPARQL queries** for further use of other related applications.

## 1.2 Motivation

The tool aims at ingesting legacy data concerning digital provenance information and data produced with tools that 3D-SYSTEK has no implementation control over them, thus cannot be integrated to the RI. It is also provided to assist the ingestion of some bulk data in order to populate early the RI. It is not provided to substitute the required integration of 3D tools with the RI.

We strongly believe that the tools integration with the RI can only be achieved by having each tool ingesting and retrieving to and from the RI, without having an intermediate tool (the “Reposit Tool”) assisting the users of each independent tool to ingest to the RI. Otherwise it is impossible to keep different metadata consistent, to manage the potential complexity of flows of work, and to keep manual interaction at the minimum necessary for industrial application.

The tool aims further at providing a good-practice library of software components for metadata creation that 3D tool providers can employ to integrate their tools more consistently and efficiently. The tool functionality of retrieving available lists of values from the RI, given specific criteria, and ingesting new such values to the RI, is provided as a set of java-functions for all other 3D tool providers. We hope that the existence of such a set of functions will help all partners to integrate their tools to the RI. We also hope that the tool source code can also be used as a coding-template for the ingestion mechanism that will be integrated in the 3D tools.

## 2 Workflow Considerations

### 2.1 Describing the output of an Event

The user normally needs to specify the output of the event which is about to describe. Actually workflow-wise this is the “first” card that the user has to fill. This is due to the fact that the user has to specify metadata for specific data located on his/her local storage: how he got these data (output), during which procedure (event), etc. We believe that this is the most intuitive way to describe an event, by assisting the user to answer the following questions: (a) which data were produced, (b) by what procedure, (c) using what input and probably (d) with some form of documentation in needed to be declared.

### 2.2 Describing details of an Event

The user describes events (either acquisition events, or process events). For each event the user should specify attributes concerning the event itself specifying:

- the title of the event,
- the time the event took place,
- the place the event took place,
- the organization,
- the operators/actors involved in the event etc.

For the description of an acquisition event the user should also specify the devices or software used for the acquisition, some information on the acquisition setup and if necessary device calibration information.

For the description of a process event the user should also specify the software used for processing their input data, the type of the software and the parameters used for the processing.

### 2.3 Describing the input of a Process Event

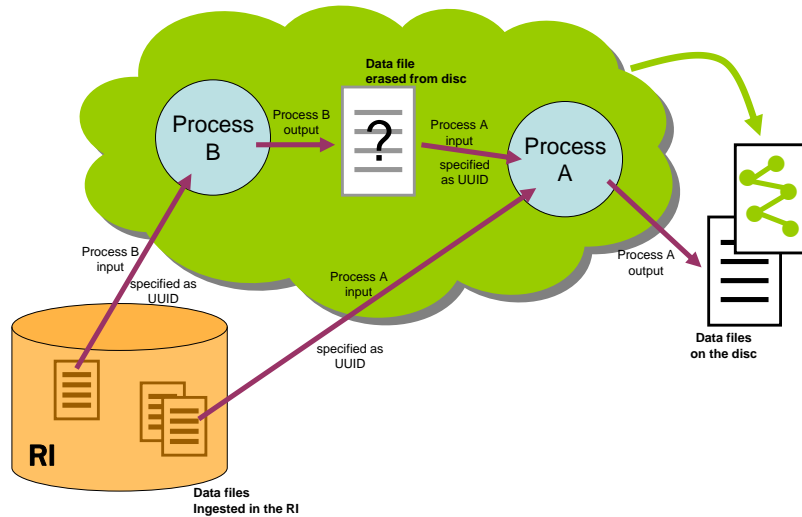
For a process event the user should also describe the input data he/she used for processing.

#### 2.3.1 All input files should be already in the RI

The tool does not allow the selection of data files that are not already ingested in the repository. This ensures data referential integrity. This also forces the user to first describe acquisition events, and the subsequent process events, which is only logical.

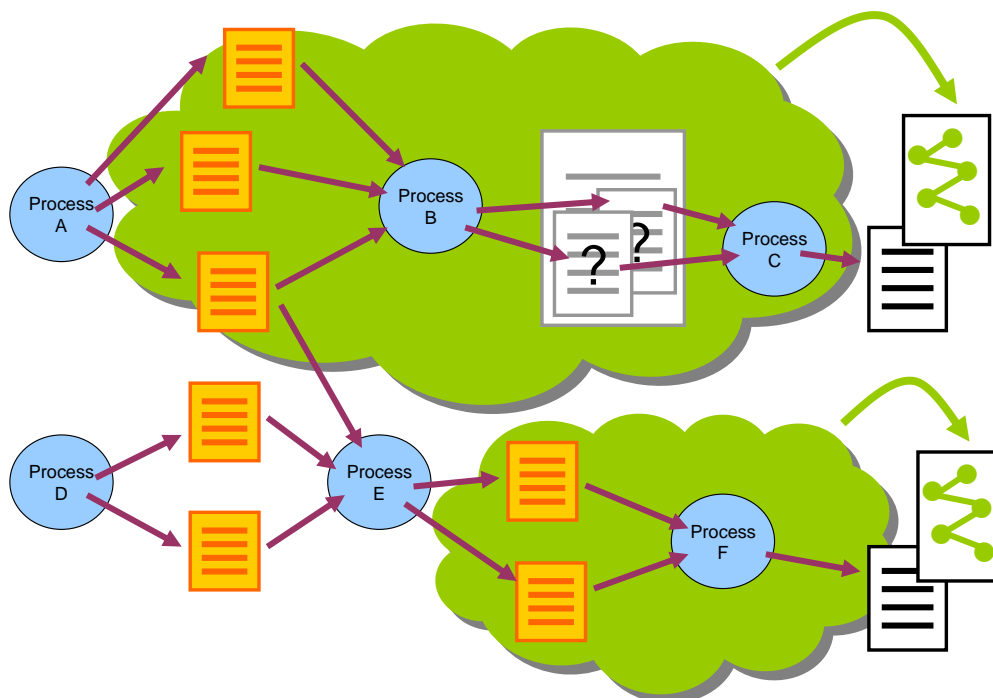
#### 2.3.2 Introduction of intermediate files

The input, except of being a set of data objects (already ingested in the repository, as mentioned earlier), can also be the product (output) of another process for which the user decided that was not needed and was erased (see Figure 1: process B output). In this case the Reposit-Tool assists the user in describing the metadata for this process too (process B), describing the software and the parameters used, up to the files used as input (already in the repository).



**Figure 1** Intermediate files may be created between two processes

This way the tool provides referential integrity when describing a chain of processes which intermediate data are erased. We will hereafter reference these products as intermediate files. The tool considers (for simplicity reasons) the intermediate files between two process as one file (see Figure 2: intermediate files between processes B and A) generating only one UUID for this intermediate data.



**Figure 2** Intermediate files between processes are considered as one file

The metadata generation in the case on intermediate files between multiple processes, include the description of all the processes up to the input files of the first process (see Figure 2 the upper green cloud denotes the generated metadata for both process C and process B, it will be ingested with the output data-file of process C).



### 2.3.3 Some process events may not have input files

There is also the case where a process does not have any input data. The user is using software to generate data files without any data files as input. The tool supports this case too by providing the user a way to declare this (lack of input).

## 2.4 Describing the documentation of an Event

In some cases we need to report information about the event in the sense of attaching of documentation files (e.g images or text files) or inserting some primitive description that summarizes the whole process. This feature could be optional for both acquisition and process event.

## 2.5 Asynchronous and synchronous ingest

The default behavior of tool is the asynchronous uploading (ingest) of files<sup>1</sup> in order to speed up the parallel ingestion of events.

However, since large files need more time for successful uploading the Reposit-Tool informs the user at the closing/exiting of the application for pending files: files that are currently uploading by the specified user during the current session (see Appendix Figure 53). The tool suggests the user to confirm that the uploads of pending files would be stopped. The user may wait until the uploading is completed<sup>2</sup>.

At the beginning of the application the tool informs the user for incomplete files that seem to be corrupted and stopped while uploading. At this case the tool informs user (see Appendix Figure 54) and applies the auto re-upload method, that re-uploads all stopped file-uploads of this user.

---

<sup>1</sup> *Synchronous ingest of files is provided as a choice through the tool.properties file.*

<sup>2</sup> At the status bar there is a link labeled as “Pending Uploads” which on click displays all the currently pending uploads per each active session. If no any pending uploads exist an appropriate message is depicted (see Appendix 55).

### 3 User Interface Description

The 3D-Systemek Reposit Tool works with the two types of events: the acquisition event and process event. It provides an interface, based on tabbed input forms, to assist the user to answer the following questions: (a) which data were produced, (b) by what procedure (c) using what input, (d) with some documentation information asserted

The purpose of the tool is to enable users to use forms instead of RDF files to interact with and be able to ingest their data in the RI. These forms interact with the RI in order to assist and assure correct input in specific fields.

The tool can save the event descriptions in files in the local disk. These files can be saved and reloaded enabling user to describe the events at his/her own time. After the description is completed the user can ingest the metadata collected along with the data these metadata describe to the RI.

#### 3.1 Login Form

If it is the first time running the tool a login form is displayed with the following 3 fields: URL (the link of repository to connect) User (username) and Password (Figure 3). After filling in this information and pressing OK the initial form of tool is displayed. The URL could be a list of URLs as denoted at the “tool.properties” file (see at Appendix). If ticked the “Remember me” option in the next run of tool these elements are restored by the system. In case of unsuccessful connection an alert message is displayed (see Figure 4).

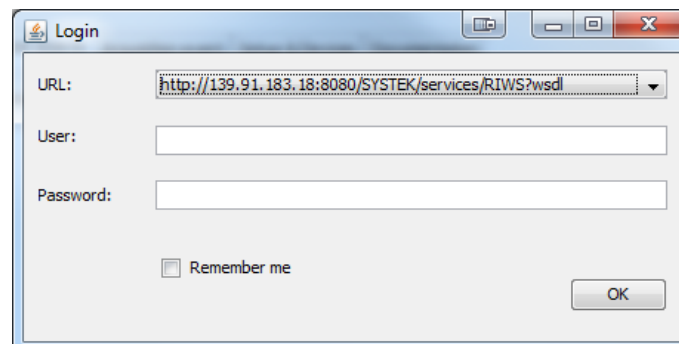


Figure 3 Login form of Reposit-tool

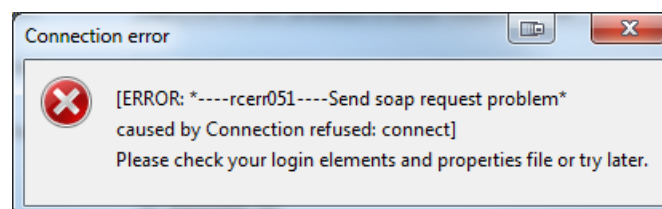
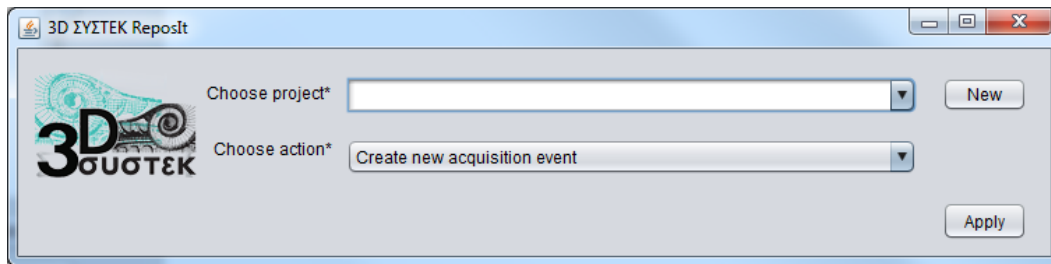


Figure 4 Unsuccessful connection error message

## 3.2 Handling an Event description



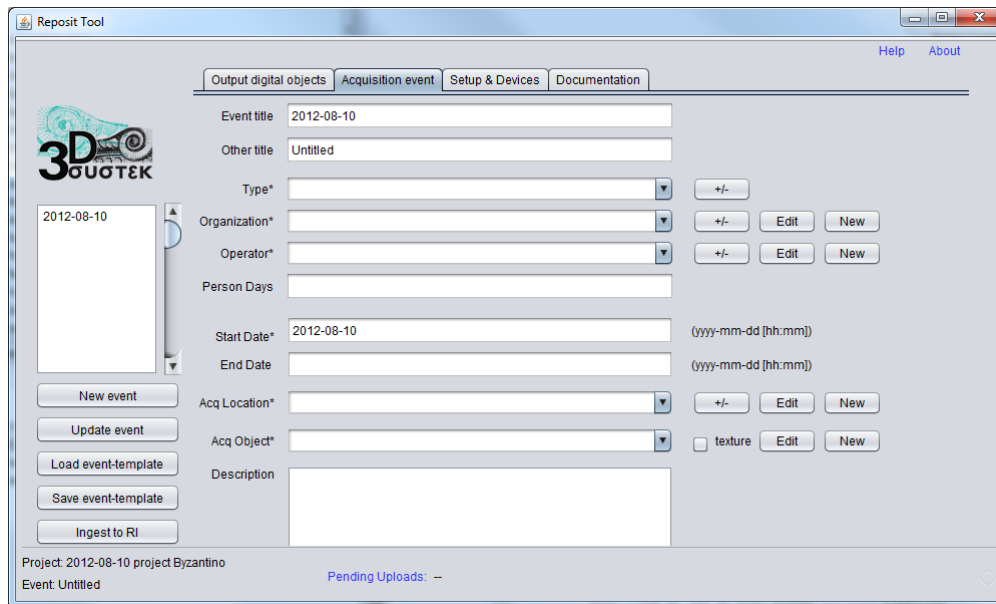
**Figure 5** Initial form of Reposit-tool

By invoking the tool the user may load an existing event description, create a new one or update an existing one. Each event is described within the framework of a “project” (see Figure 5). The user may select an existing project or create a new one by pressing the “New” button. A pop-up form for introducing the new project (see Figure 6) appears, for the user to provide all the details of the project.

**Figure 6** Adding a new Project description

In all forms at the left bottom area the following functions are available (Figure 7):

- New event
- Update Event
- Load event-template
- Save event-template
- Ingest (event) to RI



**Figure 7** From the left bottom area of main-form the user may create a new event, load and save events to disc and ingest an event

By pressing button “New event” the user may choose create a new event. If the button is pressed while editing another event the user is prompted to save or discard the current event.

By pressing button “Update event” the user may choose update an existing event. If the button is pressed while editing another event the user is prompted to save or discard the current event.

By pressing button “Load event-template” the user may choose to load an existing event description from the local storage (local hard disk).

After he/she has finished with the editing of an event he/she may save the event descriptions (for re-evaluation) in a file on the local disk (by pressing “Save event-template”), or ingest the event along with its data in the RI (by pressing “Ingest to RI”).

**Event Descriptions used as templates:** All saved event description files (once ingested) can no-longer be used for re-ingestion of the same data. Instead they may be used as “*event templates*”: they can be reloaded and used without having the user to re-enter the same information for a set of events with the same details (e.g. same organization, actors, same S/W, and S/W-parameters etc.), only having omitted all input-output information.

The left scrollbar displays the events that are currently described during one event description. Specifically during the editing of a process event the user may desire to describe more than one process events (as described in section 2.3.2): this left scrollbar enables the user to “jump” from the one process event to the other.

### 3.3 Acquisition event

The event description of an acquisition event is based on a four tabbed form: Output digital objects, Acquisition event, Setup & Devices and Documentation.

#### 3.3.1 Output Digital Object

For describing the acquisition event the user is prompted to select the output digital objects that were produced during this acquisition. This is done by pressing the “browse” button (see Figure 8). This

button pops-up a file browser, for the user, to select that data files from the local storage (local hard disk). The user is given a choice to generate a zip file (using the option “Generate zip for files”) that contains all the selected files, if the data-files to be ingested are going to be treated further as a set of data. This also reduces the metadata information that the tool generates, since it generates for one metadata file instead of many. The user is also given a choice to consider all data files as one “multipart object”, for which is prompted to define the “startup” data file. This option is mutually exclusive to the option of “generate zip for files” (see Figure 8).

When updating an acquisition event all forms described below are already filled by the event details retrieved from the RI. Filenames of digital objects or other files that are already ingested in the RI are denoted by a “#” preceding the filename.

The screenshot shows the 'Reposit Tool' window with the 'Acquisition event' tab selected. On the left, there is a sidebar with a '3D ΣΥΣΤΕΚ' logo and a list of events, with '2012-08-10 non port' selected. Below the list are buttons for 'New event', 'Update event', 'Load event-template', 'Save event-template', and 'Ingest to RI'. The main area has a 'Filename' list with three entries: 'mnimeio1.jpg', 'mnimeio2.jpg', and 'mnimeio\_original.jpg'. Below the list are two checkboxes: 'Generate zip for files' (unchecked) and 'Consider all output objects as multipart with startup object' (checked). A dropdown menu below the second checkbox shows 'mnimeio\_original.jpg'. At the bottom, the project name is '2012-08-10 project Byzantino' and the event name is 'Untitled'. There is also a 'Pending Uploads' indicator.

**Figure 8** Form for selecting the output of an acquisition event

For each of the displayed digital objects the user could optionally store individual metadata information by selecting a file in the list (see Figure 8). Then a new form pops-up (see Figure 9). This edit form enables the user to add a note to the selected digital object and set (or update) its thumbnail image. For process event user can select a specified type (e.g 2D image, video etc). If option “Generate zip for files”) is selected then the metadata information and thumbnail on each individual file is ignored. If in update mode the selection “to be deleted form the RI” is available, for the user to declare that the digital object is no longer part of the specified event.

The screenshot shows the 'Digital object details' form. It has several input fields: 'ID' with the value 'uuid:026d8505-8a34-4e7e-9f0e-63cf31b71090', 'File' with 'mnimeio2.jpg', and 'Type' with a dropdown menu set to 'image/jpeg'. There is a 'Note' text area, a 'Thumbnail' text area, and a 'Browse' button next to it. At the bottom right, there is an 'Apply' button.

**Figure 9** Form for editing output file attributes

### 3.3.2 Acquisition Event

By clicking at the tab “Acquisition event” the form describing basic information for this acquisition event appears (see Figure 10). The user can insert information such as the title of the event, the time and the place the event took place, the responsible organization, and the operators involved in the event etc.

**Figure 10** Form for editing the description and details of an acquisition event

The user can specify a value in a field (e.g. Operator) by selecting a value from a drop-down list, of values from the RI. This list also enables type-ahead and use of sub-strings for narrowing the selection of available values, from the drop-down. Type-ahead search activated after typing at least three characters.

Fields with “\*” on their labels are obligatory.

Fields in grey are not editable. These may be fields that their value is calculated by the values of other fields. Such fields are the Event title fields, e.g an acquisition event title is calculated by the Event type, date, museum object, etc. This calculation mechanism may be necessary to enable automatic translation and multilingual search capability of some fields.

The user can also create a new entity (such as a new Person) in the RI, by “New” button, or update an existing entity in the RI, by “Edit” button. A pop-up form appears (see Figure 11) for introducing the new entity (or for updating the existing one), for the user to provide all the details of the entity. In Figure 11 is presented the form for creating/updating a Person identifying first and last name of the person, the organization the person is currently a member of, his role in the organization, a unique URI (optional) for identifying the Person entity (if not given the system will create a UUID for this entity) and a note field. In section 3.5 are presented all the forms for creating new Entities.

The “+/-” button is designed to enable the user to specify more than one values in specific input field (see section 3.6.2)].

The screenshot shows a window titled "Create/update person". It contains the following fields and controls:

- First name\* (text input)
- Last name\* (text input)
- URI (text input)
- Member of\* (dropdown menu)
- Role (dropdown menu)
- Note (text area)
- Buttons: Edit, New, and Apply.

**Figure 11** Form for creating a Person

The “Acq Location” field specifies the place the Acquisition took place. The form for introducing a new acquisition location provides the user the choice of specifying locations (see Figure 12), as specific as a room in a building of an organization (see Figure 13), or a set of coordinates (see Figure 14) in the case of an outdoor acquisition. In the last case a link to geonames website is provided for convenience.

The screenshot shows a window titled "Choose type for acquisition location". It contains:

- Location Type: (dropdown menu showing "Building/Exhibition Space")
- Apply (button)

**Figure 12** Form for creating a Location

The screenshot shows a window titled "Create/update building or exhibition space". It contains the following fields and controls:

- Building/Exhibition space URI (text input)
- Building/Exhibition space name\* (text input)
- Place\* (dropdown menu)
- Address\* (text input)
- Buttons: Edit, New, and Apply.

**Figure 13** Form for creating a Building/Exhibition Space

The screenshot shows a window titled "create/update Coordinates". It contains the following fields and controls:

- Location URI (text input) with a "Geonames" link to its right.
- Location Title\* (text input)
- Longitude (x)\* (text input)
- Latitude (y)\* (text input)
- Place\* (dropdown menu)
- Address\* (text input)
- Buttons: Edit, New, and Apply.

**Figure 14** Form for creating a Location by coordinates

The “Acq Object” field designates the physical object. In Figure 15 is presented the form for specifying the object that is being digitized. The user can specify whether is a museum object, or an immobile object or site (e.g. archeological site, public heritage monument, sculpture or building). The fields in the form varies according the user selection: in Figure 15 are presented the fields for specifying a museum object by its inventory number, title, type, otherID, size (relative), keeper (museum) and note. No other form of legacy data is ingested by this tool.

**Figure 15** Form for identifying a Museum Object

### 3.3.3 Setup & Devices

By clicking at the tab “Setup & Devices” the form for describing information about the hardware device data (device, characteristics and note) used for this acquisition event appears (see Figure 16). In this form the user can also specify other devices used, such as lenses, flashes, etc.

The tool provides more than one ways to describe the setup configuration: by a plain text filed (field “Setup Characteristics”) or by one or more configuration-files. The form provides a field to specify the calibration files used for this acquisition, if any (field “Calibration file”).



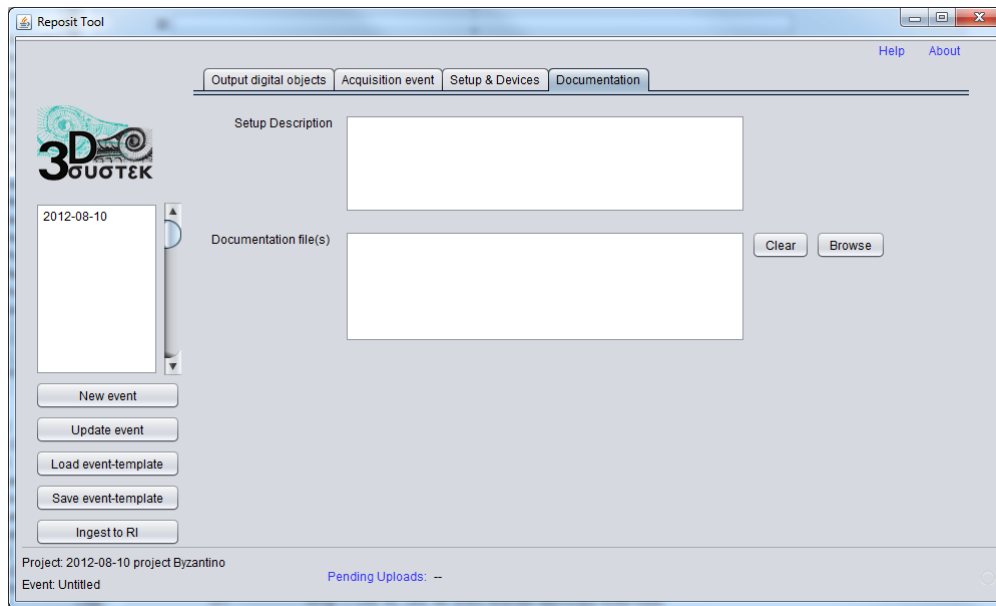
**Figure 16** Form for editing the setup and devices of an acquisition event

In Figure 17 is presented the form for creating a Device entity, defining details like device-type, serial number, maker, firmware, notes etc. This form also provides the user with a sub-form for describing details on the documentation concerning the acquisition setup (button “Setup Documentation”).

**Figure 17** Form for creating a device

### 3.3.4 Documentation

In documentation tab the user can describe in a field the setup of the whole process and include (ingest) related documentation files for example photos/video of the setup and possible documents with details on the setup installation by clicking the button “Browse” (see Figure 18).



**Figure 18** Documentation tab for acquisition event

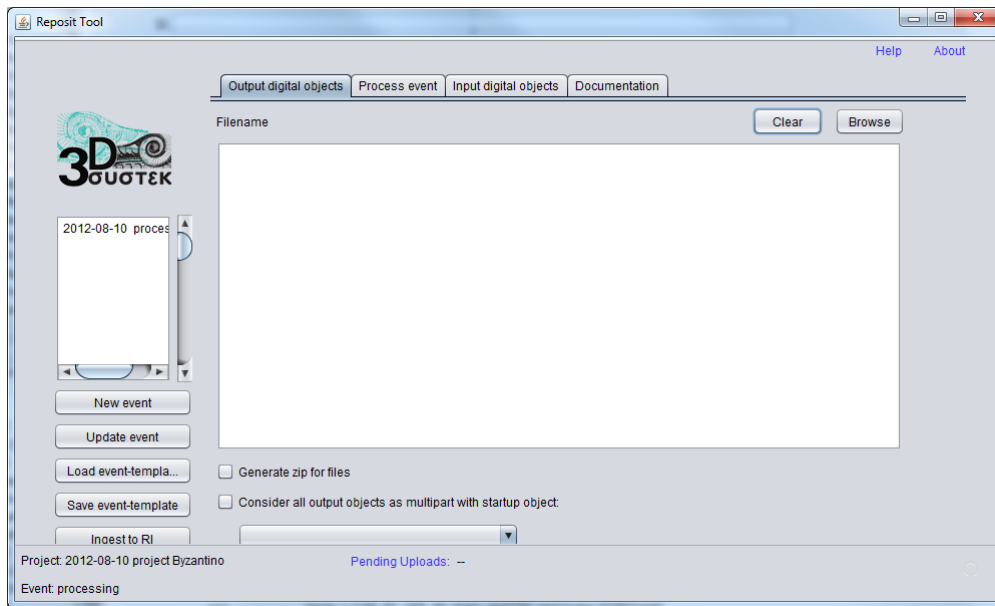
## 3.4 Process Event

The event description of a process event is based also on a four tabbed form: Output digital objects, Process event, and Input digital objects, Documentation.

### 3.4.1 Output Digital Object

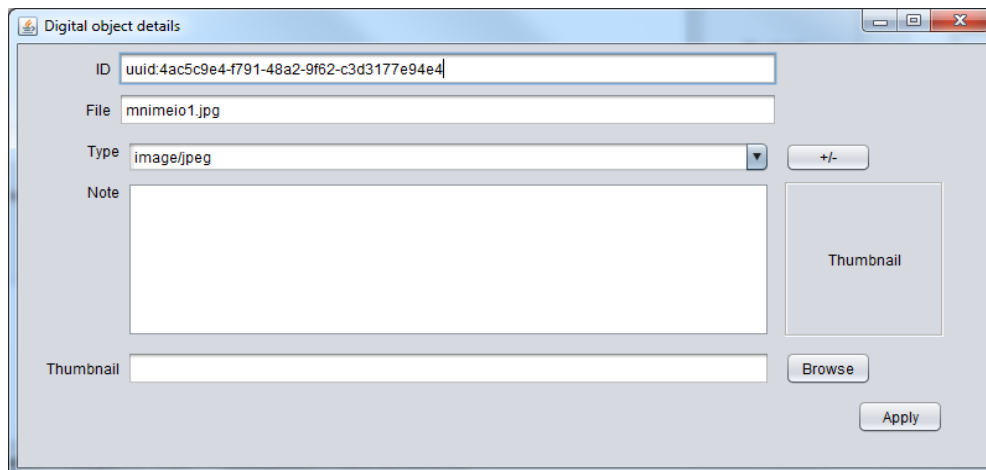
Similarly to acquisition event the user is prompted to select the output digital objects that were produced during this process event. This is done by pressing the “browse” button (see Figure 19). This button pops-up a file browser, for the user, to select that data files from the local storage (local hard disk). The user is given a choice to generate a zip file (using the option “Generate zip for files”) that contains all the selected files if the data-files to be ingested are going to be treated further as a set of data. This also reduces the metadata information that the tool generates, since it generates for one file instead of many. The user is also given a choice to consider all data files as one “multipart object”, for which is prompted to define the “startup” data file. This option is mutually exclusive to the option of “generate zip for files”.

When updating a process event all forms described below are already filled by the event details retrieved from the RI. Filenames of digital objects or other files that are already ingested in the RI are denoted by a “#” preceding the filename.



**Figure 19** Form for selecting the output of a process event

For each of the displayed digital objects the user could store individual metadata information by selecting a file in the list. Then a new pop-up form appears (see Figure 21). This edit form enables the user to add a note to the selected digital object and set (or update) its thumbnail image. If option “Generate zip for files”) is selected then the metadata information and thumbnail on each individual file is ignored. If in update mode the selection “to be deleted form the RI” is available via “Clear” button, for the user to declare that the digital object is no longer part of the specified event.



**Figure 20** Form for editing output file attributes

### 3.4.2 Process Event

By clicking at the tab “Process event” the form describing basic information for this process event appears (see Figure 21). The user can insert information such as the title of the event, the time the event took place, the organization, and the operators/actors involved in the event, the software used for processing, the parameters used for the processing, etc.

Note that the tool provides more than one ways to describe the parameters: by a plain text field (field “Software Parameters”) or by one or more parameter-file.

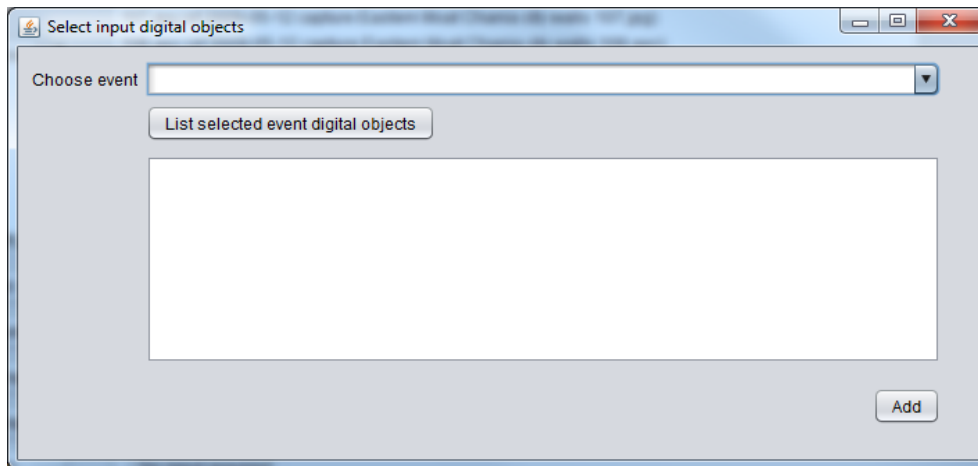
**Figure 21** Form for editing the description and details of a process event

### 3.4.3 Input Digital Objects

For describing the input of the process event, the user can select digital objects from RI (see Figure 22). By pressing “Query” button a pop-up form appears (see Figure 23) to enable the user to select as input, the output files of a “previous” event: files that were already been ingested (maybe using the Reposit-Tool at a previous point in time).

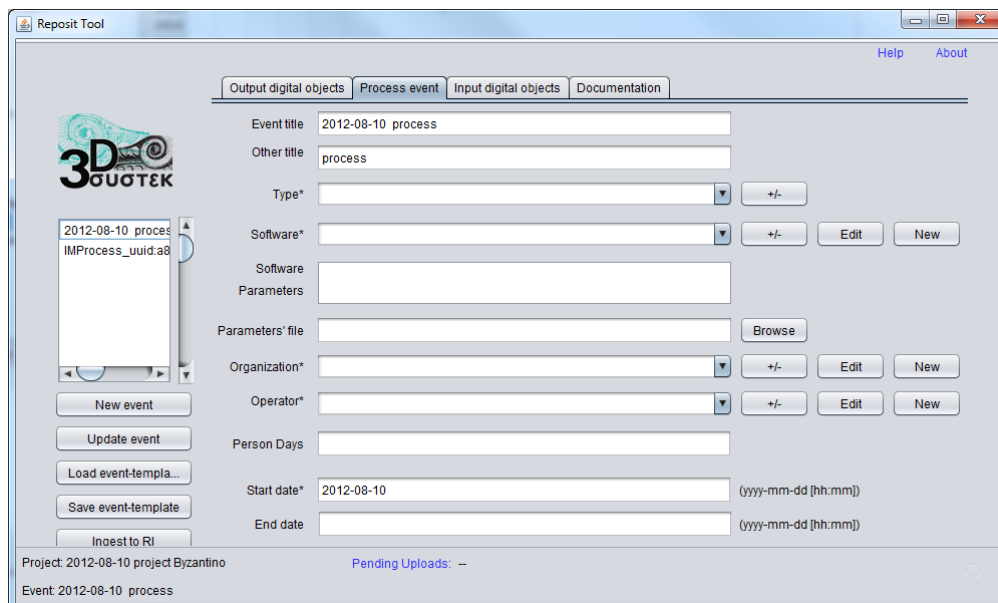
The process input, except of being a set of data objects from the RI, can also be the product (output) of another process for which the user decided that was not needed and has erased (as described in section 2.3). In this case the Reposit-Tool assists the user in describing the metadata for this process, creating and describing a new process (by pressing “Describe earlier intermediate process event” button). These choices of input can be supported both for the same process event. On the other hand if the user clicks on the checkbox “No input required”, then the above input selections are disabled and their contents are ignored.

**Figure 22** Form for selecting the input (file-uids or process output) of a process event



**Figure 23** Form for selecting the output files of a “previous” event.

By pressing the “Describe earlier intermediate process event” button the user can enter information for the new process (see Figure 24) and choose the corresponding input digital objects (see Figure 25) with the same procedure as above.



**Figure 24** Form for editing the description and details of the process that produces the intermediate files

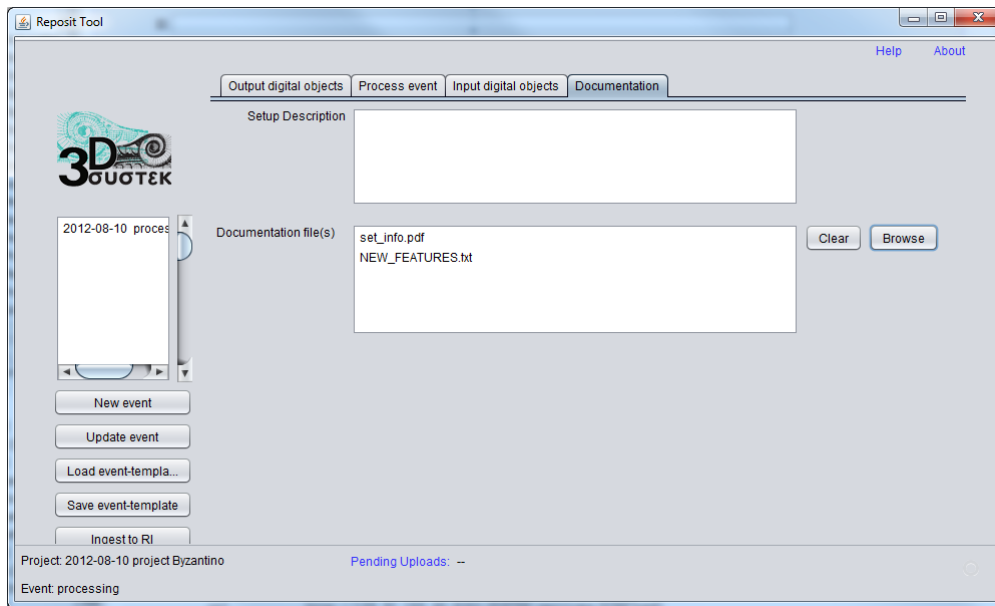
**Figure 25** Form for selecting the input of the intermediate files generating process

The left scrollbar displays the events that are currently described during one event description, enabling the user to “jump” from the one process event to the other. Thus (as shown in Figure 26) once the user has finished with describing the new process, can go back and review or continue the description of the first process, by selecting the process name/title on the left scrollbar.

**Figure 26** Updated form with selected input (file-uuids or process output) of a process event

### 3.4.4 Documentation

In documentation tab the user can describe in a field the setup of the whole process and include (ingest) related documentation files for example photos/video of the setup and possible documents with details on the setup installation by clicking the button “Browse” (see Figure 27).



**Figure 27** Documentation tab for process event

### 3.5 Forms for the creation and update of entities

The user can also create new entities (such as a new Person, new Legal Body, new Software, new Place, etc.) in the RI, by “New” button, or update existing entities in the RI, by “Edit” button, in forms described above. A pop-up form appears for introducing the new entity (or for updating the existing one), for the user to provide all the details of the entity (see figures: Figure 28, Figure 29, Figure 30, Figure 31, Figure 32, Figure 33, Figure 34, Figure 35).

**Figure 28** Form for creating/updating a Person

**Figure 29** Form for creating/updating a Legal Body

**Figure 30** Form for creating a Place

Examples of geonames ID (or GeoNameId) For the town *Athens* in Greece we have these two URIs :

- [1] <http://sws.geonames.org/264371/>
- [2] <http://sws.geonames.org/264371/about.rdf>

How to find a PLACE To find the GeoNameId (through the geonames link) of a place you can issue the following query, substituting PLACE with the place-name you are looking for (e.g. Athens):

<http://ws.geonames.org/search?type=rdf&q=PLACE>

**Figure 31** Form for creating/updating a Building/Exhibition Space

**Figure 32** Form for creating/updating a Location by coordinates



The screenshot shows a window titled "Create/edit software". It contains the following fields and controls:

- Software URI: Text input field.
- Name: Text input field.
- Manufacturer\*: Dropdown menu with a +/- button to its right.
- Type\*: Dropdown menu with a +/- button to its right.
- Note: Large text area.
- Action buttons: Edit, New, and Apply.

**Figure 33** Form for editing software

The screenshot shows a window titled "Create/edit digital device". It contains the following fields and controls:

- Digital device ID: Text input field.
- Digital device name\*: Text input field.
- Type\*: Dropdown menu with a +/- button to its right.
- Serial number\*: Text input field.
- Maker\*: Dropdown menu with Edit and New buttons to its right.
- Device characteristics: Large text area.
- Firmware: Dropdown menu with Edit and New buttons to its right.
- Action buttons: Apply.

**Figure 34** Form for creating/updating a Digital Device

The screenshot shows a window titled "Create/edit physical object". It contains the following fields and controls:

- Physical object ID: Text input field.
- Title\*: Text input field.
- Object name: Text input field.
- Inventory no: Text input field.
- Other ID: Text input field.
- Size: Dropdown menu.
- Type\*: Dropdown menu with a +/- button to its right.
- Keeper\*: Dropdown menu with Edit and New buttons to its right.
- Note: Large text area.
- Action buttons: Apply.

**Figure 35** Form for creating/updating a Physical Object

## 3.6 Form Input fields functionality

The forms of the Reposit-tool have input fields like those shown in Figure 38.

### 3.6.1 Input field functionality

The user can either select a value from the drop-down list or start typing in and the RI is queried for similar strings of the specific type. Type-ahead in input-fields narrows the selection of available values, from the drop-down list. By typing in the input field, the available values that match the typed string are narrowed (see Figure 36). *Important notice:* due to extremely long lists of available values the type-ahead mechanism is enabled after typing at-least three characters. This mechanism uses Lucene engine for optimized queries.

**Figure 36** Input field type-ahead use example

Notice that fields in grey are not editable and that fields with “\*” on their labels are obligatory.

### 3.6.2 Multi-valued Input fields

In some input fields the user can specify more than one values. By selecting “+/-” button a form will appear (see Figure 37). In this form the user may add more than one value (more than one Organization, Person, etc).

**Figure 37** Multi-valued Input field use example

### 3.6.3 Creation and update of entities

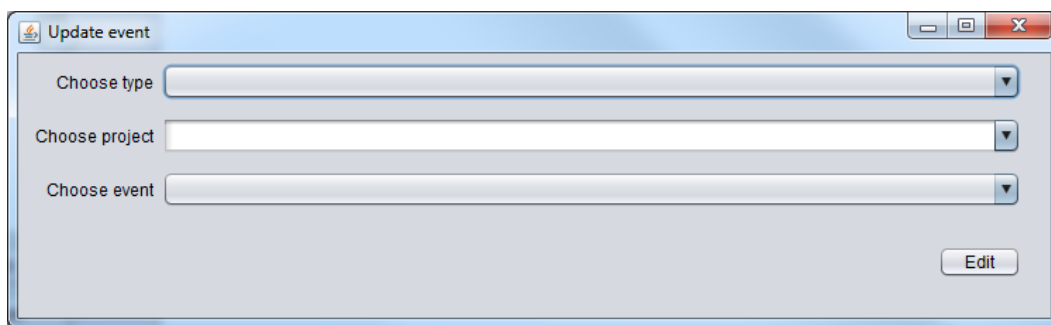
The user can also insert new entities (such as new Persons) or update entities (only those created using the Reposit-Tool), by selecting “New” or “Edit” button respectively (see Figure 38). Then a form for introducing the new entity pops-up to provide all the details of the entity.

**Figure 38** Input fields examples

It should be noted that the displayed value in a field, should be compact, in order to be easily selectable, thus it may vary from the detailed information presented when a new entity is introduced. See section 3.5 for a presentation of the available forms for creating entities.

### 3.6.4 Update of Events

The tool also permits the editing and updating of existing Event-metadata (already ingested in the RI), provided they comply with the Reposit-Tool disciplines. The user may select “Update event” button and a pop-up will appear, which allows selecting an Event, from a list of Events (see Figure 39). Specifically the user first choose type of event (first combo) and optionally a specified project (second combo) and finally selects from the third combo box a specified event to be updated. The user can change any field of each event (output, input, and other fields) and press “Ingest to RI” button for activating the update process.



**Figure 39** Form for selecting an Event for update.

### 3.6.5 Creation of new Types

To create new type-values in a existing vocabulary such new acquisition types, new device types, etc please contact [georgis@ics.forth.gr](mailto:georgis@ics.forth.gr) or [hrysakis@ics.forth.gr](mailto:hrysakis@ics.forth.gr) requesting the creation of the new type.

## 4 Appendix

### 4.1 Tool.properties file

The tool.properties files is used as configuration file which loads at each run at the beginning of the Reposit-Tool. Below we present a common example of this file.

The **property "urls"** specifies the URLs (comma separated URLs) of the RI to be connected to (e.g. <http://testserver/SYSTEK/ObjectRepository.svc>,<http://139.91.183.45:8080/SYSTEK/services/RIWS?wsdl>).

The property **"deletefiles"** if set to no allows the user to "review" the generated RDF metadata files. The property instructs the tool not to delete the generated RDF files after ingestion completion. These files are located in the RDFS folder of the tool.

The property **"querylog"** if set to yes allows the user to "review" all executed queries, their results and their execution time and is used for debugging purposes. All this information captured in a file named queries.txt inside the RDFS folder.

The property **"sync"** if set to yes enables synchronous uploading of data and metadata.

Also the user has the ability to initialize entity combos by setting corresponding variables in the properties file, with a specific regular expression pattern for each specific entity combo. The regular expression pattern syntax follows the java-regex format (for more details see <http://docs.oracle.com/javase/7/docs/api/java/util/regex/Pattern.html>) and is case-insensitive.

The values of the following example properties file are identical to the default ones. That means that if we don't set them or omit some of them, the combos would be initialized by these default values. The last can be overwritten by setting different values in the properties file. For example by setting `LegalBodyInitEntities = ^a`, all combos related with Legal Bodies should be initially loaded by Legal Bodies which label starts with a.

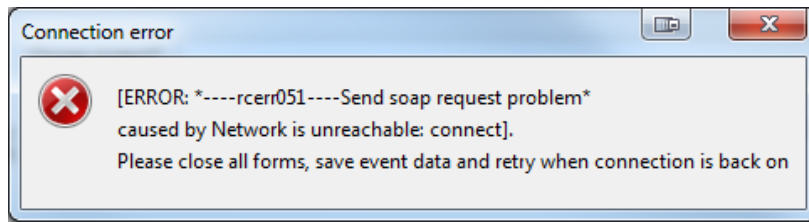
```
# Reposit Tool Properties File

urls = http://139.91.183.45:8080/SYSTEK/services/RIWS?wsdl
deletefiles = no
querylog = yes
sync = no

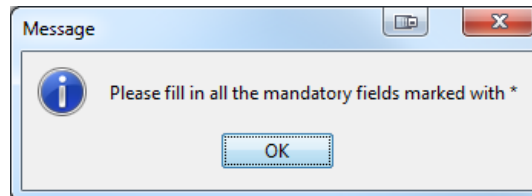
# optional initialization for combos (java regular expressions)
PersonInitEntities = ^[abcd]
LegalBodyInitEntities = ^[abcd]
LocationInitEntities = ^[abcd]
PhysicalObjectInitEntities = ^[abcd]
DigitalDeviceInitEntities = ^[abcd]
SoftwareInitEntities = ^[abcd]
EventInitEntities = ^20
PhysicalObjectTypes = ^a
VocabularyTypeItems = ^.
```

### 4.2 User Control Messages

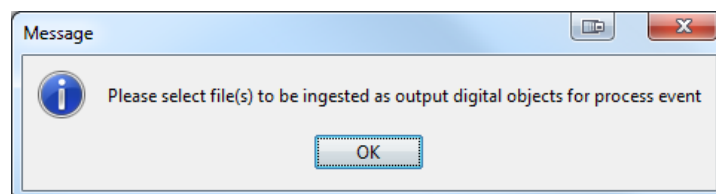
We present a list of basic messages that are displaying due to control operations of Reposit-Tool's user.



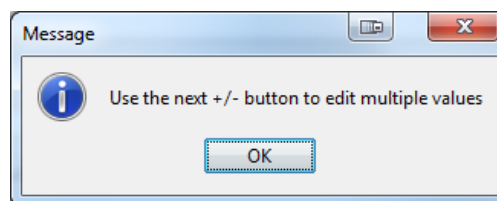
**Figure 40** Message for connection error



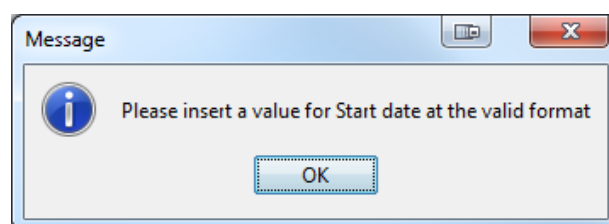
**Figure 41** Message for completing mandatory fields



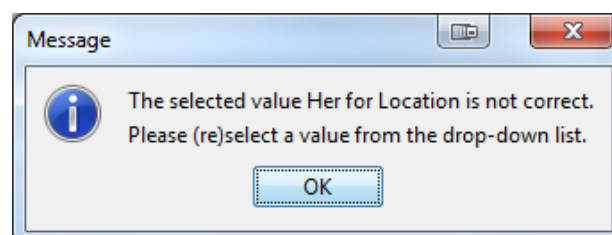
**Figure 42** Message for missing output digital object at event



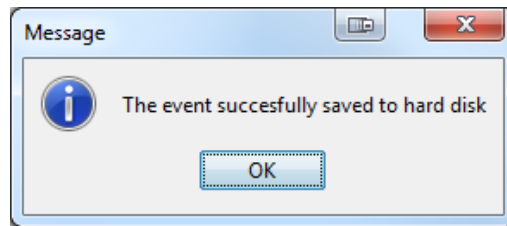
**Figure 43** Message for editing multiple values



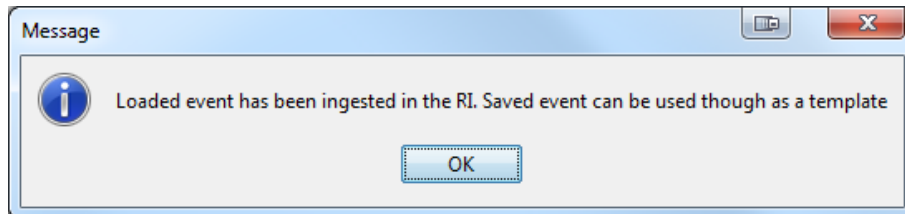
**Figure 44** Message for wrong date format



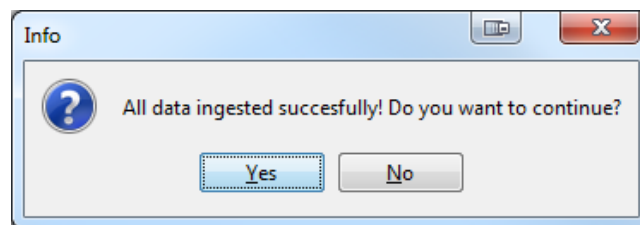
**Figure 45** Message to re-select a value from a specified combo box



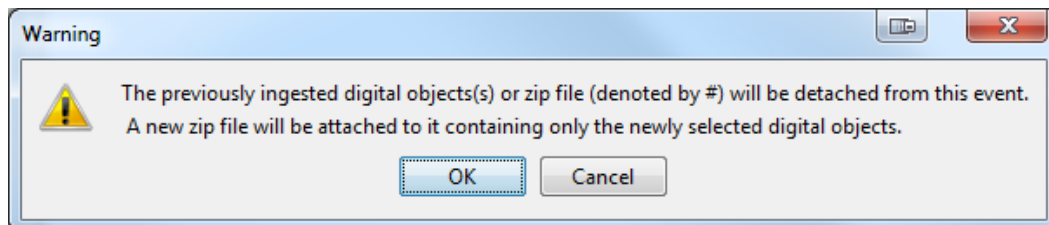
**Figure 46** Message for successful save of an event



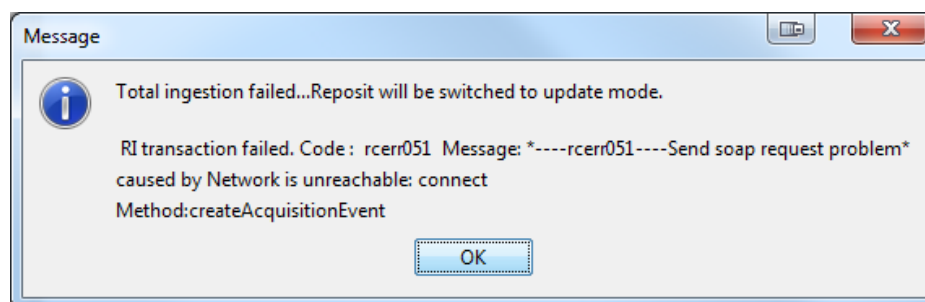
**Figure 47** Message for loading an event as template



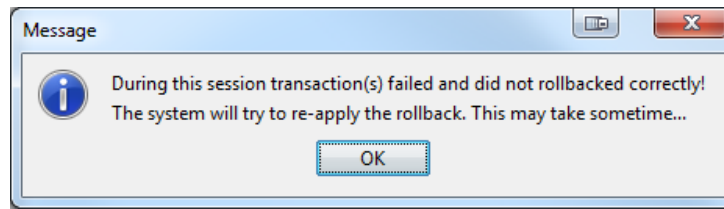
**Figure 48** Message for successful ingestion



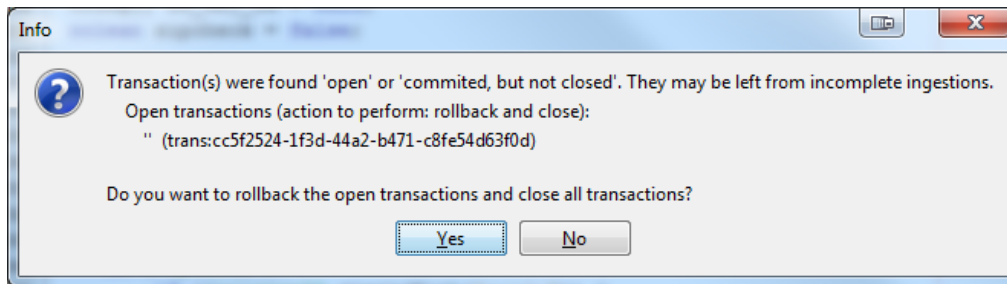
**Figure 49** Message for detaching a zip file from an existing event



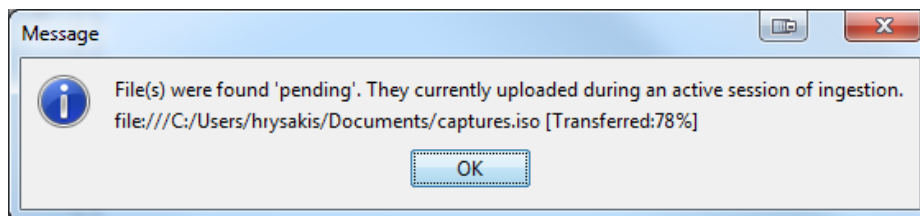
**Figure 50** Message for auto-switching to update mode in case of failure as soon as the main information (metadata) of event has been ingested



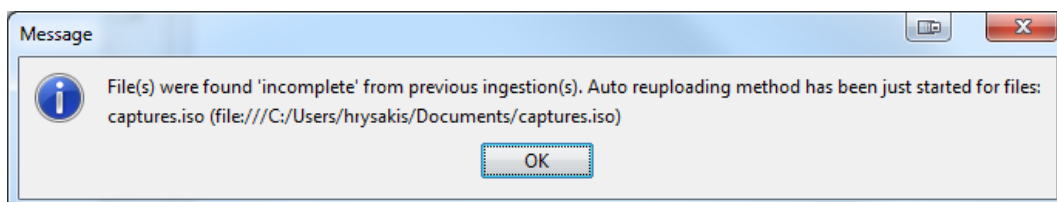
**Figure 51** Information message for rollback mechanism which has been enabled by a transaction failure



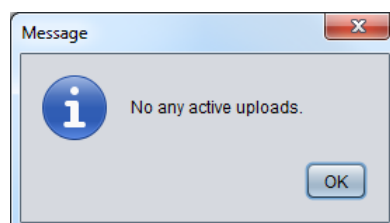
**Figure 52** Information message for open-transactions with option to apply rollback



**Figure 53** Message at closing of application to inform for active uploads during this session (pending files).



**Figure 54** Message at start of application to inform for incomplete uploads (incomplete files). The system tries to re-upload these files from their original destinations.



**Figure 55** Message which is displayed when clicked at the status bar and no any pending uploads exist.