Chapter 6

Importing Data

EAD 2002

The Archivists' Toolkit™ allows for the importing of valid EAD version 2002 finding aids into description resource and component records.

> Note: The AT does not guarantee complete round-tripping of data (i.e. the data that is imported into the system may not be exactly the same as the data that is exported). For more details, see the section on data mapping at the end of this chapter, and the EAD to AT data map in the appendices.

Constraints on EAD

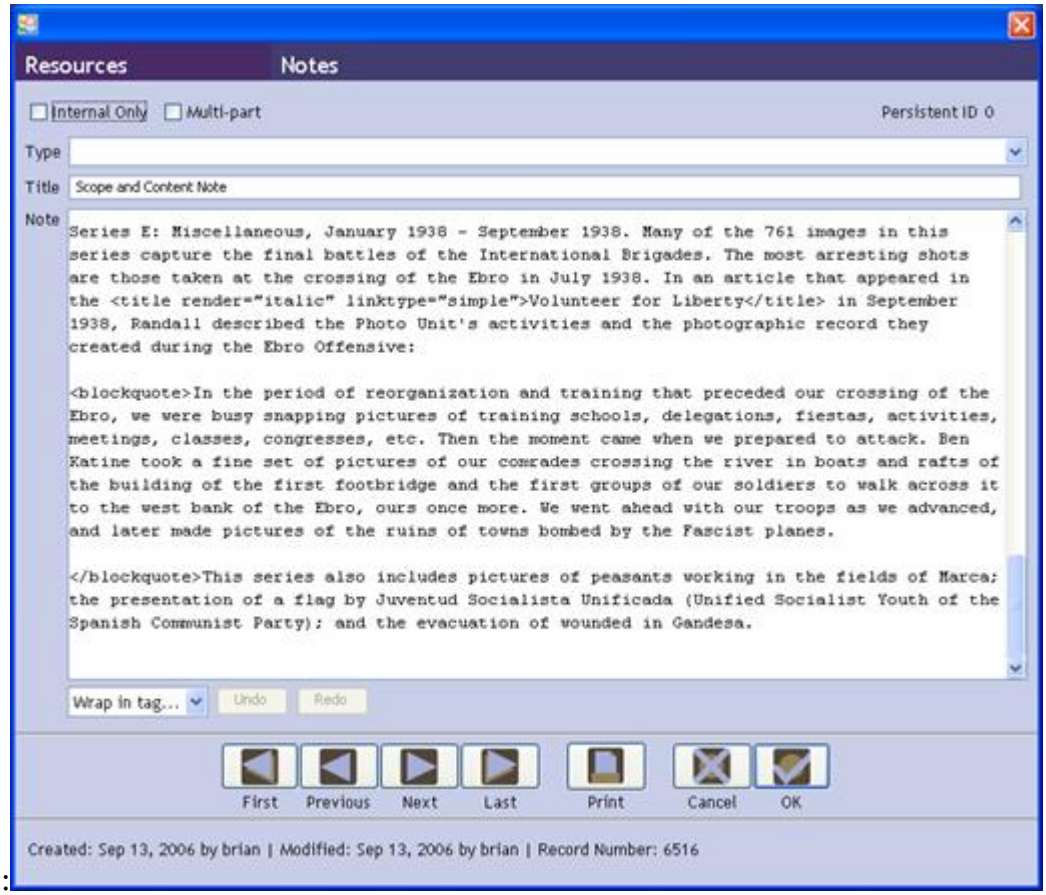
To be imported, the EAD must meet the following conditions:

1. EAD files must:
    a. Be valid version 2002 documents. Version 1.0 EAD files need to be converted to version 2002 prior to import.
    b. Contain a <unitid> within the <archdesc>.
    c. Conform to the EAD DTD or to the EAD schema (EADs using the EAD group DTD are not accommodated in the AT EAD import process).
2. Not all valid EAD tagging can be accommodated by the Toolkit's data model, though such instances are rare. Where inline tagging is encountered in a source EAD, it is imported into the Toolkit as mixed content and is visible as XML in its appropriate database field. Examples of inline content include formatting tags (such as <emph>, <lb/> and tables), access points outside of <controlaccess> tags, and external references (with the exception of dao types, which are mapped as digital instances). For example, a note encoded as follows:

```
. . .taken at the crossing of the Ebro in July 1938. In an
article that appeared in the <title
render="italic">Volunteer for Liberty</title> in September
. . . Ebro Offensive:</p>

<blockquote><p>

In the period of reorganization ...

</p></blockquote>
```

Will be imported with some tagging retained as mixed content

:



Note: As seen in the example above, paragraphs are displayed in the AT with two line returns. Therefore, the <p> tags do not appear in the note.

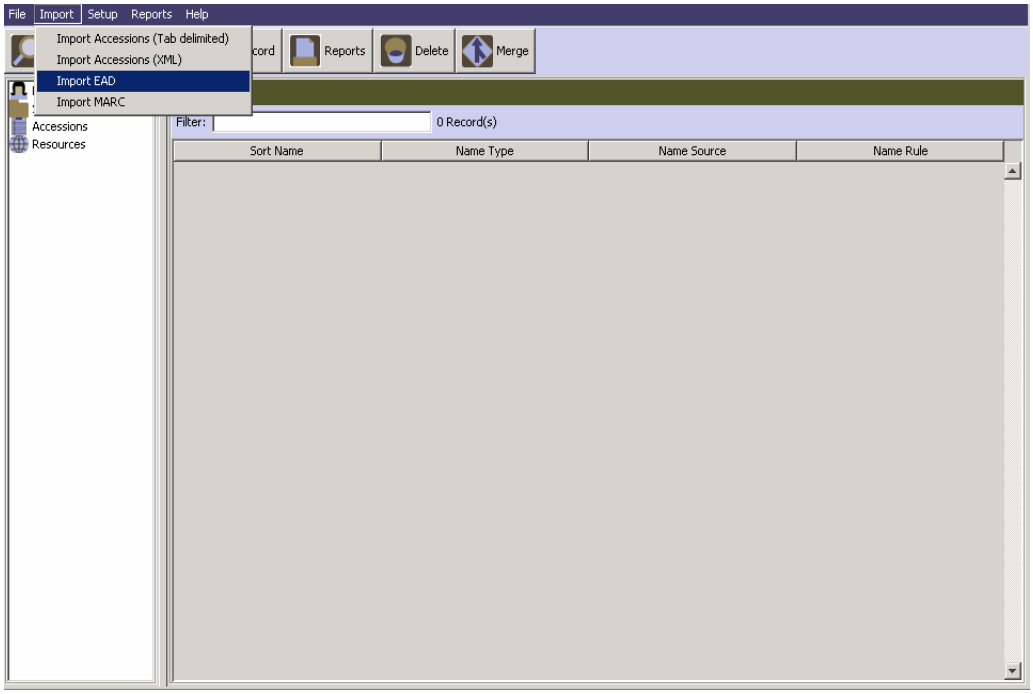
4. Most internal referencing between id and target tag attributes are retained with some provisions: the Toolkit's "persistent identifier" replaces the id target pairing and the Toolkit's "targets" are limited to notes and components.

5. The Toolkit does not currently import a digital object **title** or **objectType** for <dao> elements in EADs even though these fields are required in the Toolkit. If you are importing EADs containing <dao> elements, you will need to add **title** and **objectType** information to these records in the Toolkit so that the resulting digital object records are valid. (It is possible, however, to export these as <dao> elements in EAD even if the digital object record is invalid in the Toolkit.)

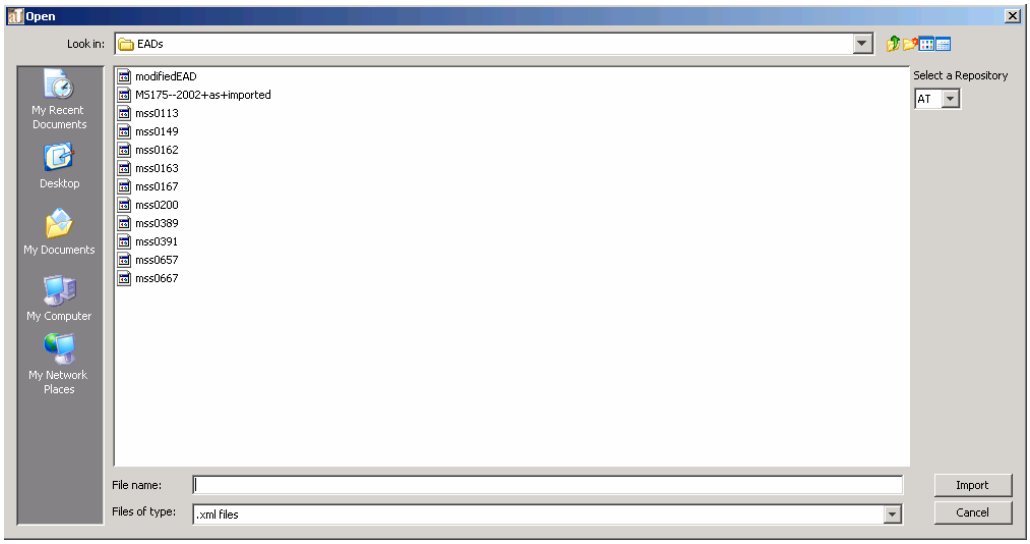
6. Named entities are not supported. Entities must be resolved before import. For example, if you use entities to reference special characters, substitute the Unicode hexadecimal character for the named entity.

Importing a single EAD file

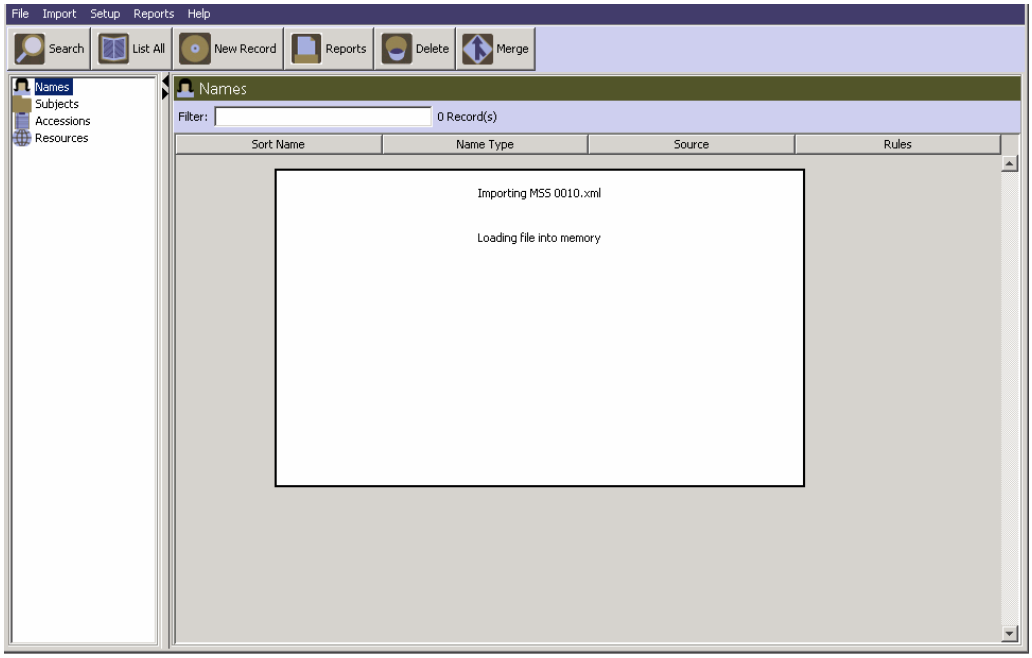
1. From the **Import** menu, select **Import EAD**.



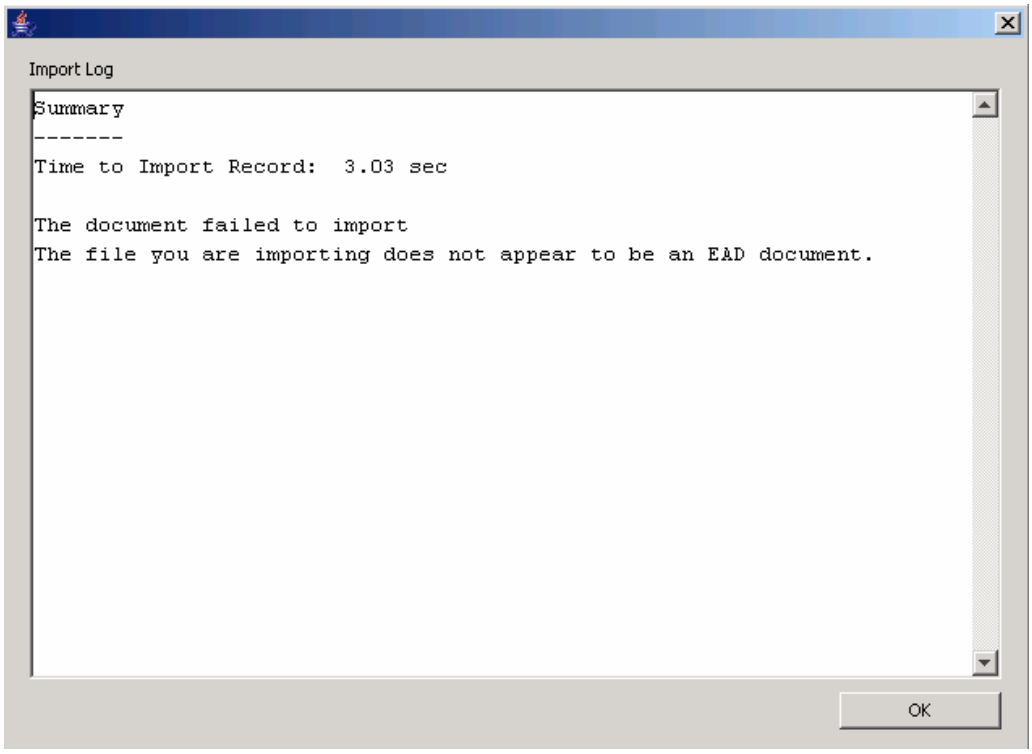
2. Choose the file to be imported.
3. Select the repository that holds the materials described in the finding aid.
4. Press the Import button.



The Toolkit will display a message indicating the progress of the import:



If an error is encountered, such as an invalid EAD, the Toolkit will display an error message indicating that the file could not be imported:

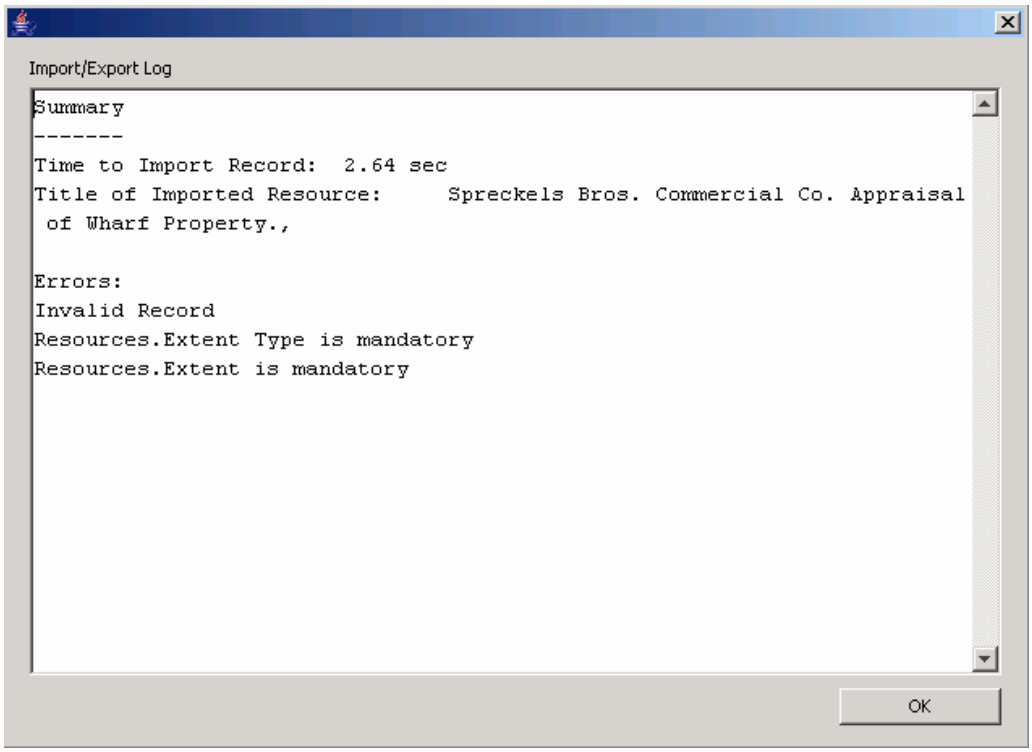


There are several factors that could result in a failed import of an EAD instance:

- The EAD instance is not a version 2002 EAD.
- The EAD instance is not well formed: all start tags need to have end tags, all elements need to be correctly nested, all files should have a root element, and attribute values should be enclosed in quotation marks.

- The EAD to be imported already exists in the Toolkit as a multi-level resource, that is, there is already a resource record in the Toolkit with the same resource identifier as the EAD to be imported and that resource record has linked resource component records. If the resource record does not have linked resource component records, then the EAD selected for import will be imported and will overwrite or merge with fields in the existing Toolkit resource record. Fields that are present on re-import will overwrite the existing field, and fields that are not present on re-import will retain the old values rather than inserting a blank or null value for that field. The problem may be corrected by deleting the resource description already in the AT and then importing again the EAD that failed to import. It can also be avoided by restricting editing processes to within the Toolkit.
- There should be no line breaks in the <ead> tag, including breaks in between attributes of the tag. Otherwise the EAD will not import.
- Named entities must be resolved before import.

If import of the record is successful, the **Import Log** appears, and lists any record validation errors in the resulting AT record(s):

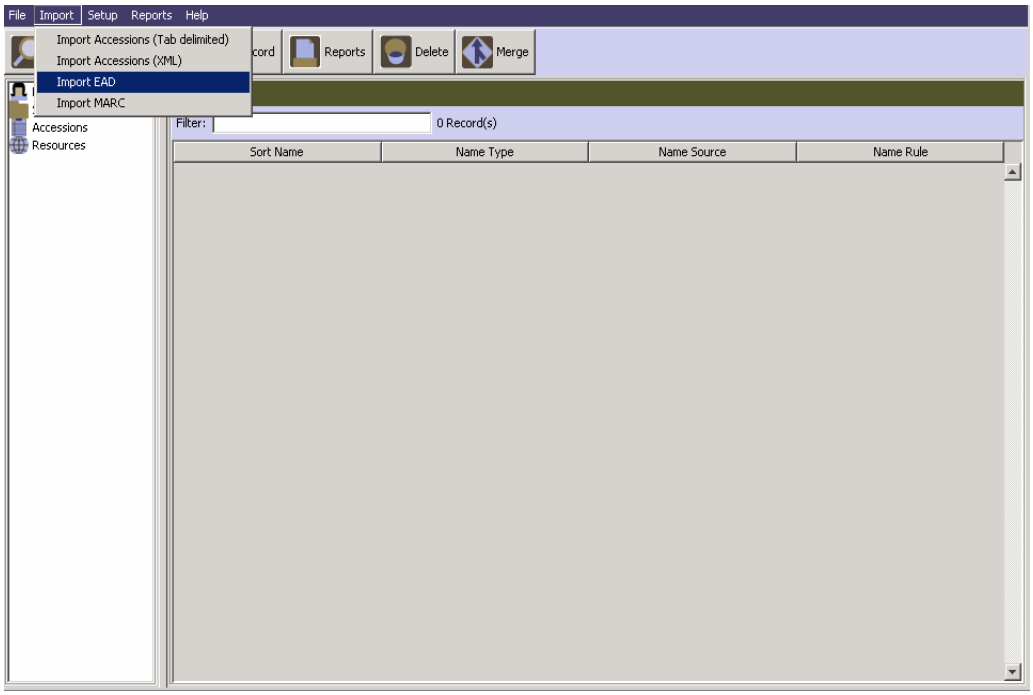


These particular errors indicate how the newly created AT record fails to meet AT record requirements. They typically consist of absent data elements that are necessary for a valid AT record. You will be prompted to correct these errors once the resulting record is opened in the AT. They must be corrected in order for you to save the record again.

In addition, the **Import** log will list items that have been added to lookup lists. These items should be reviewed for accuracy and in most instances removed as they can affect system functionality. More information about data cleanup is listed at the end of this chapter.

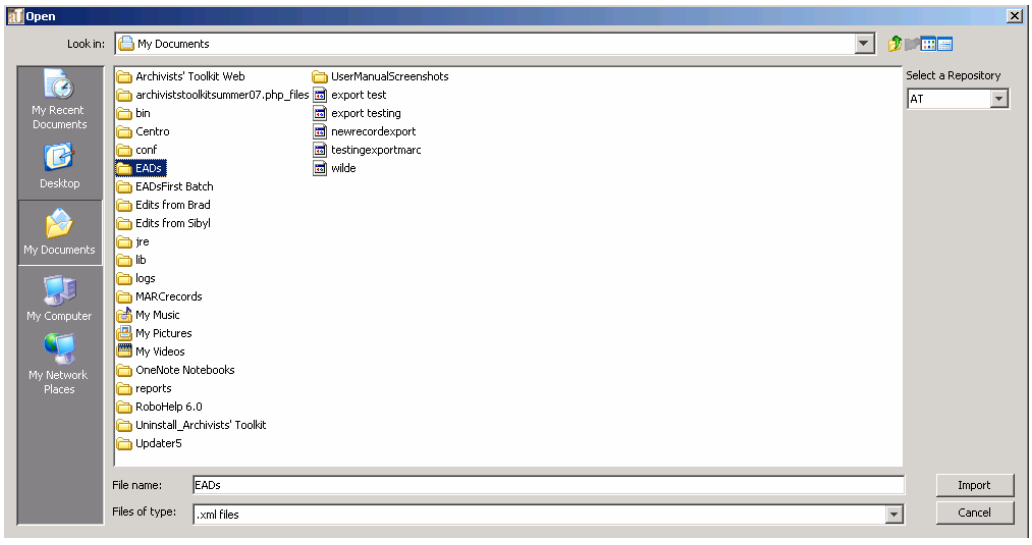
Batch Importing EAD

1. From the **Import** menu, select **Import EAD.**



2. Choose the folder that contains the EAD files to be imported.

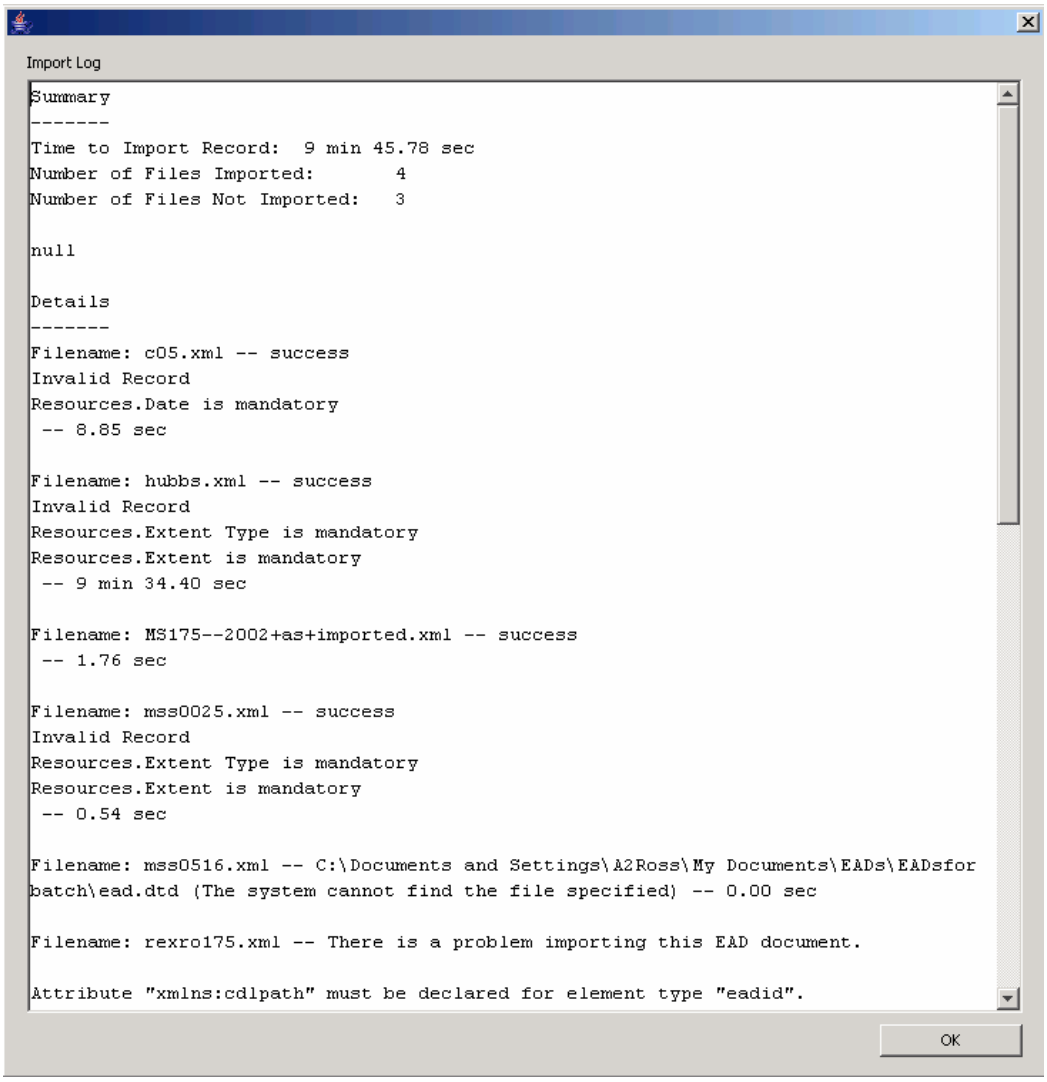
3. Select the repository that holds the materials described in the finding aid. Only one repository can be selected. If you are importing EAD files from various repositories, the files should be grouped by repository and imported in smaller batches.



4. Press the Import button.

5. The Toolkit will proceed in the same manner it does when importing a single record; it will first display a processing message, and then either output a failed import message if the import is unsuccessful, or output an import log. See the section on importing a single EAD record for troubleshooting failed import messages.

6. If the import is at least partially successful (i.e. some of the EADs were imported), the **Import Log** will list any errors that occurred during the import process. These errors typically consist of absent data elements that are necessary for a valid AT record. You will be prompted to correct these errors once you open the resulting record in the AT. They must be corrected in order for you to save the record again. The import log will also display a file-by-file listing of which EADs were successfully imported.



Mapping of EAD elements to Archivists' Toolkit™ fields

The entire EAD data structure does not map directly to the AT's data structure; in order to make sure that no data is lost, certain elements are imported as mixed content. That is, some data will be imported into a field with all of its internal tagging intact. Tables and linking elements are examples of EAD elements that are imported as mixed content. For example, a biographical note (<bioghist>) that contains a <table> will be imported into the Toolkit's **Biographical/ historical note** field, with all content and tagging.

Note: A complete EAD to AT mapping is available in the appendices. It is organized into four sections, each on an individual tab in the Excel spreadsheet. Though some aspects of the mapping are explained below, those interested in

the complete details of how EAD elements are imported into the Toolkit should refer to the mapping document.

Linking elements are also imported as mixed content. For internal references imported into the Toolkit (<ptr>,<ptrloc>, <ref>, <refloc>) the target and id pairs are realigned so that they contain Toolkit persistent ID values. For descriptions created within the AT, only <extref> and <ref> elements are used by the **Wrap in tag** editor, though you are free to hand code any other linking elements you wish to use.

A final note on linking elements relates to importing indexes; within an <index>, the Toolkit only supports one link per index entry. Therefore, <ptrgrp> elements are not imported.

Due to the flexibility of EAD, some instances of imported data will not map to the desired field. For example, the complete content of <revisiondesc> is mapped in the AT to the **Revision Description** field. The <date> attribute is not parsed into the **Revision Date** field because there can be multiple <revisiondesc><date> elements in the imported EAD. It is necessary to manually place the <date> attribute into the **Revision Date** field for the EAD to export correctly. If this is not done, the <revisiondesc> element will not be present in the EAD export. This is because a date element is required for a valid EAD <revisiondesc><change> element.

On import, data in the <daodesc> is mapped to a general note, with the title of "Digital Archival Object Description" attached to the newly created digital object. Data is imported to this location (rather than to a more specific data element) because usage of the <daodesc> varies widely. The data as imported into general note will not be included in EAD exports generated from the parent resource; the newly exported <daodesc> will be populated with data from the digital objects title and date fields. Exports (MODS, DC, MARCXML) of the digital object record will include the general note (according to the export data maps), and so post-import clean up of the data from <daodesc> is recommended. Depending on the nature of information, this can be done by moving it to a more appropriate element or deleting it from the record.

MARCXML

The Archivists' Toolkit™ allows you to import valid MARCXML records into description resource records; however, the AT is not designed to manage MARC records. The MARC import function is provided simply to allow repositories to bring all their resources within a single management tool, whether they are described in legacy MARCXML records, EAD records, or created from scratch in the Toolkit.

Caution! The AT MARC import function does not support round tripping of MARCXML records (i.e., the MARCXML record exported from the AT will not be the same as the MARCXML record that was imported). The import of MARCXML records into the AT involves loss of data granularity, and in some cases, loss of data. For instance, some subfield demarcations are not supported in the AT so that although the data is imported, it is concatenated within the same AT field. While the AT is designed to support the MARCXML data elements typically used for describing archival materials, there may be cases where an imported MARCXML record contains a field not supported by the AT.

Despite these constraints, repositories may wish to import MARCXML in the following scenarios:

- When legacy data for archival resources is in the form of MARCXML records and not in the form of EAD finding aids and the repository wants to manage the resources described in the MARCXML records in the AT.
- When the MARCXML records for archival materials are considered to contain the only, or the more authoritative, name and subject headings than what are present in a repository's EAD finding aids.

Import has been intentionally designed as a "one record at a time" process based on the aforementioned recommendations for MARCXML import scenarios. Though MARCXML records usually are exported from an Integrated Library System as a "batch," importing batches of resources may cause AT records to be unintentionally overwritten or duplicated.

Constraints on MARCXML

To be imported, the MARCXML record must meet the following conditions:

1. It must be a valid MARCXML record with type coded to "bibliographic."
2. The source file must contain only a single MARCXML record.
3. Top level MARCXML records that contain a 773 field, indicating the record is a "child" record related to a "parent" record, will not be imported.
4. The MARCXML record must contain a 210, 222, 240, or 245 title field or the record will not be imported.
5. If the tagging follows the <marc:[tag]> convention, the namespace declaration of the imported document must be:

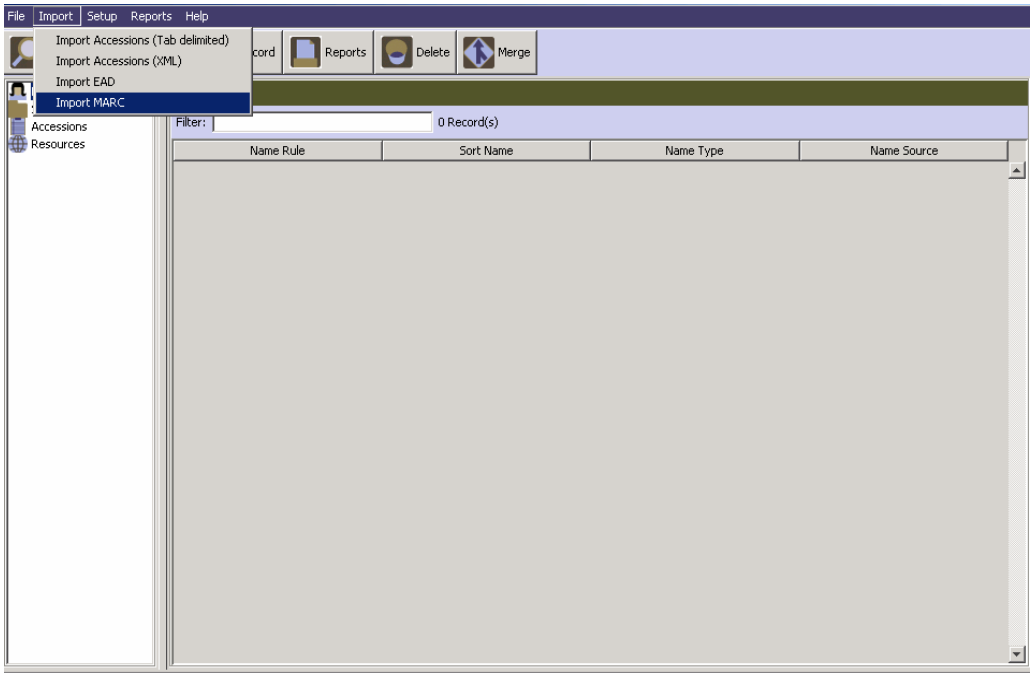
```
xmlns:marc="http://www.loc.gov/MARC21/slim"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation=http://www.loc.gov/MARC21/slim
http://www.loc.gov/standards/marcxml/schema/MARC21slim.xsd
```

Otherwise the namespace declaration must be:

```
xmlns="http://www.loc.gov/MARC21/slim"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.loc.gov/MARC21/slim
http://www.loc.gov/standards/marcxml/schema/MARC21.xsd"
```

Importing MARCXML

1. From the **Setup** menu, select **Import,** then **Import MARC**.

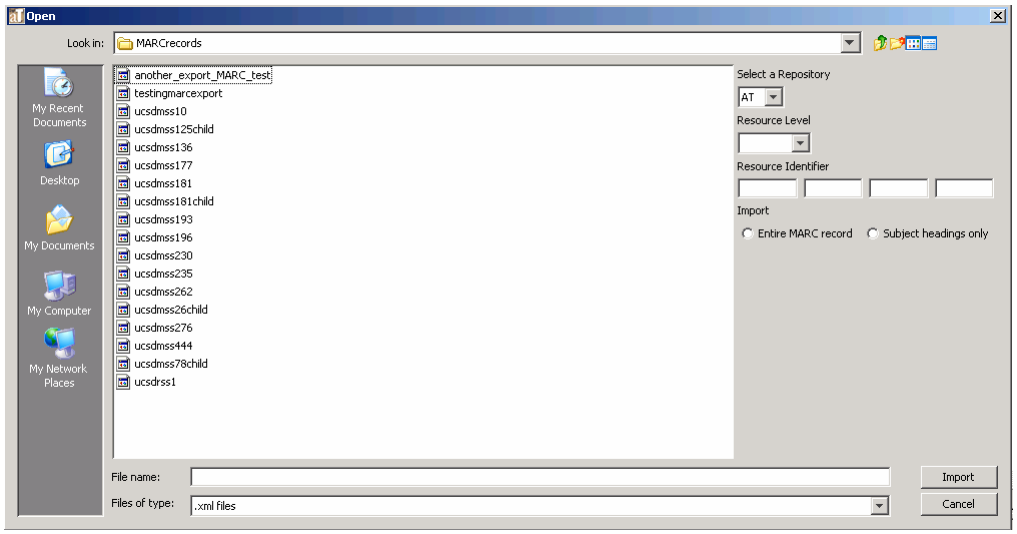


2. Choose the file containing the MARCXML record to import.
3. Select the repository that holds the materials described in the MARCXML record.
4. Indicate resource identifier.

Note: If the resource identifier matches the resource identifier of an existing record, the Toolkit will ascertain whether or not the existing AT record contains linked component records. If it does not contain linked component records, the resource record will be overwritten. If it does contain linked component records, the Toolkit will respond that the MARCXML record cannot be imported.

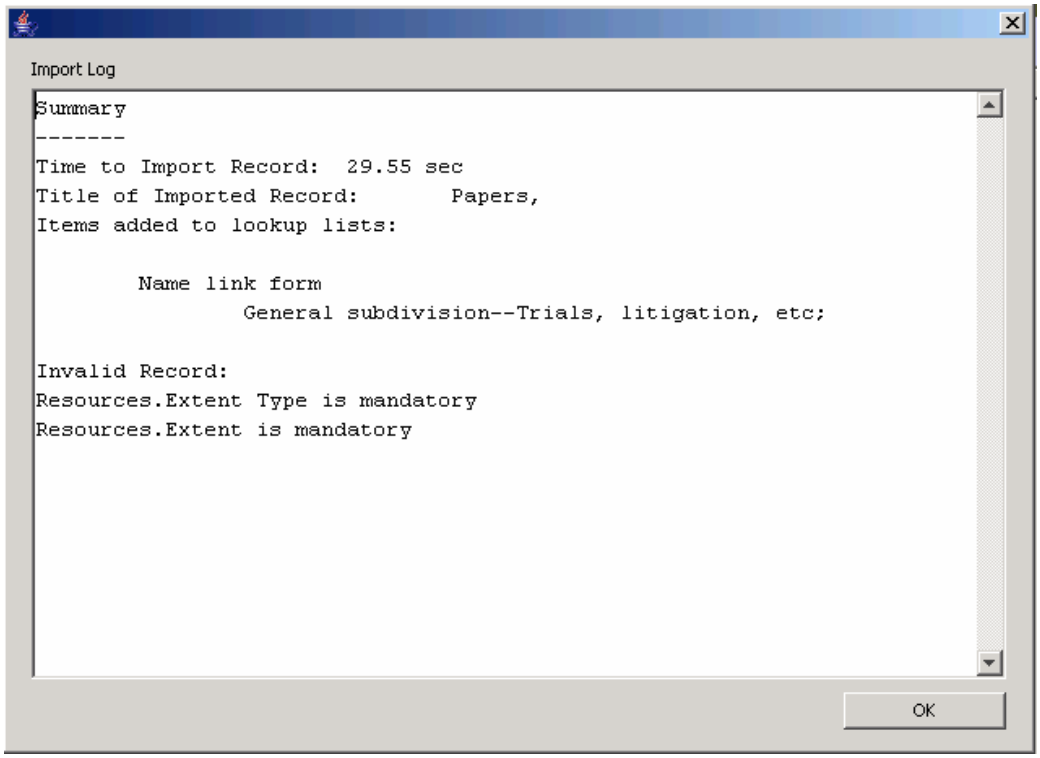
---

5. Indicate whether you wish to import the entire MARCXML record or ONLY the 1xx, 6xx, and 7xx fields for name and subject headings.



6. Press the Import button.

If an error, such as an invalid MARCXML is encountered, the Toolkit will display an error message indicating that the file could not be imported. If import of the record is successful, an import log will appear, listing any validation errors present in the imported record:



These errors typically consist of absent data elements that are necessary for a valid AT record. The first time you open the resulting AT record, the AT will prompt you to correct these errors, and will not allow you to save the record again until the corrections have been made.

Mapping of MARCXML elements to Archivists' Toolkit™ fields

A MARCXML to AT mapping is available in the Data Maps section of the appendices.

Tab-delimited accessions data

Importing accessions data

The Archivists Toolkit™ allows you to import data from either a tab-delimited file from a table or flat file, or from an XML file structured according to the Accessions XML schema provided with the Toolkit. This data may be imported into the accessions, names, or subjects functional areas. There are certain limitations inherent in implementing the tab-delimited method: only one name and subject type can be imported per accession. The tab-delimited import is also more likely to introduce data errors that result in extra data clean-up work. The XML accessions schema was designed to address the limitations of the tab-delimited ingest method. While the XML schema method is recommended for

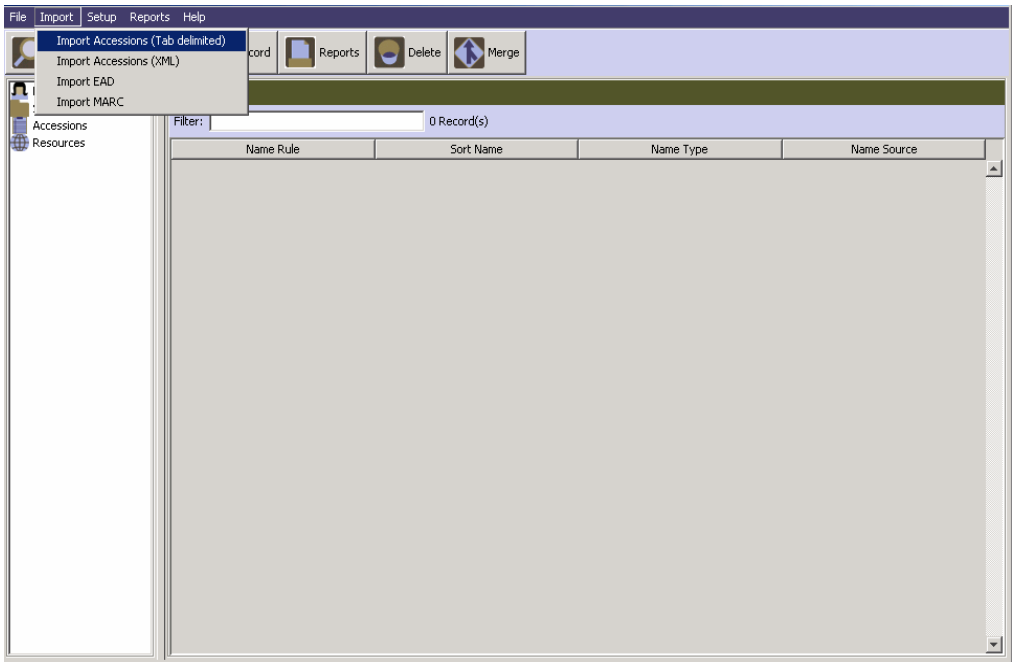
these reasons, it does require more technical knowledge in order to transfer data from a database to the XML format.

Importing tab-delimited accessions data

The Toolkit will not import information straight from a database, such as Access or Filemaker Pro. To transfer accessions data to the Toolkit, a tab-delimited file must be exported from the database containing the information. This tab-delimited file can then be imported into the Toolkit. Before importing data into the Toolkit, the data must be mapped to the fields in the Toolkit. See the **Preparing the Tab-delimited Accession File** appendix for mapping instructions.

Importing accessions data

1. From the **Import** menu, select **Import Accessions (Tab delimited)**



2. Make the following selections:

    a. Choose the file to be imported.

    b. Select the repository to which the accession data applies.

    c. Indicate the format of the dates contained in the file to be imported.

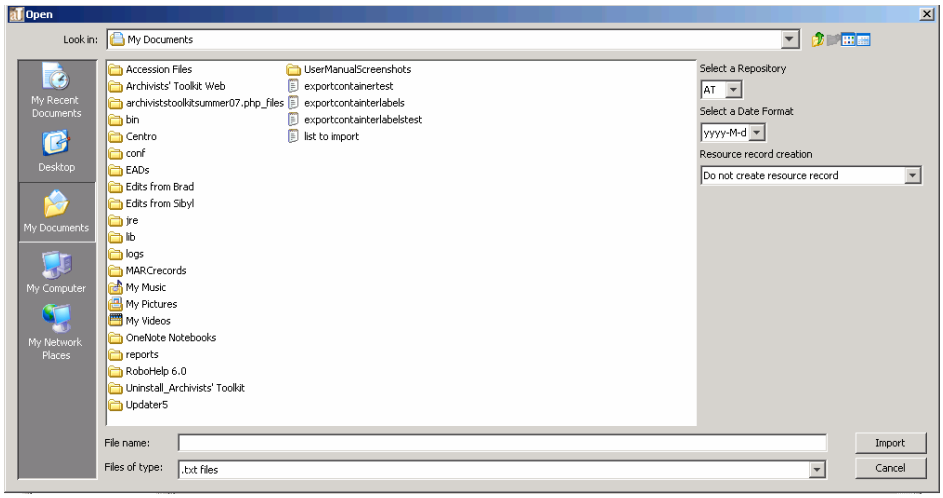
    d. Select the appropriate **Resource record creation** option.

    Do not create resource record. No resource records will be created; only accession records. If there is a **resource identifier** in the import file the system will check to see if the resource exists and link to it.

    Create resource with resource id only. If there is a **resource identifier** in the import file the system will check to see if the resource exists and link to it. If not, a new

resource record will be created. Only the **resource identifier** and **repository** fields will be present in the resource record; all other fields will be empty.

Create resource record using all fields. If there is a **resource identifier** in the import file the system will check to see if the resource exists and link to it. If not, a new resource record will be created. All of the fields that can be transferred from an accession record will be populated in the resource record. See Chapter 7 for a table listing how these fields are mapped.



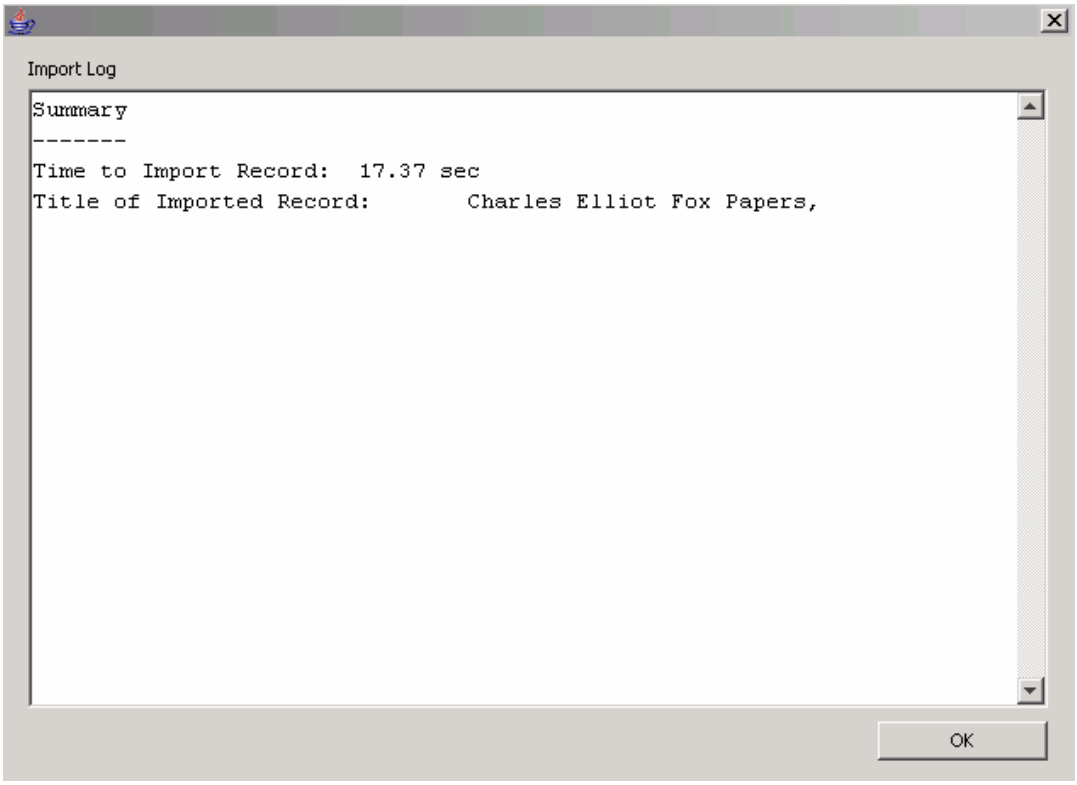
3. Press the Import button to begin.

   A progress window will track the import process. If errors are encountered, an error log will appear at the end of the process.

Note: Several error conditions can cause the import to fail, in whole or in part:
- An improperly formatted import document will cause the entire process to fail. No records will be imported.
- An invalid record will not be imported. To be valid an accession record must have an accession number and an accession date. A list of validation rules is provided in the appendices.
- Incorrectly formatted data, e.g., a non-accepted date format, or data mismatch, e.g., text data where integer data is required, will cause import of a record to fail.

If the import document is formatted correctly, the process will proceed record by record. All valid and error-free records will be imported. All invalid and / or errant records will not be imported and will be listed as such in the resulting log.

If no errors are encountered, you will see a window like the one shown below.



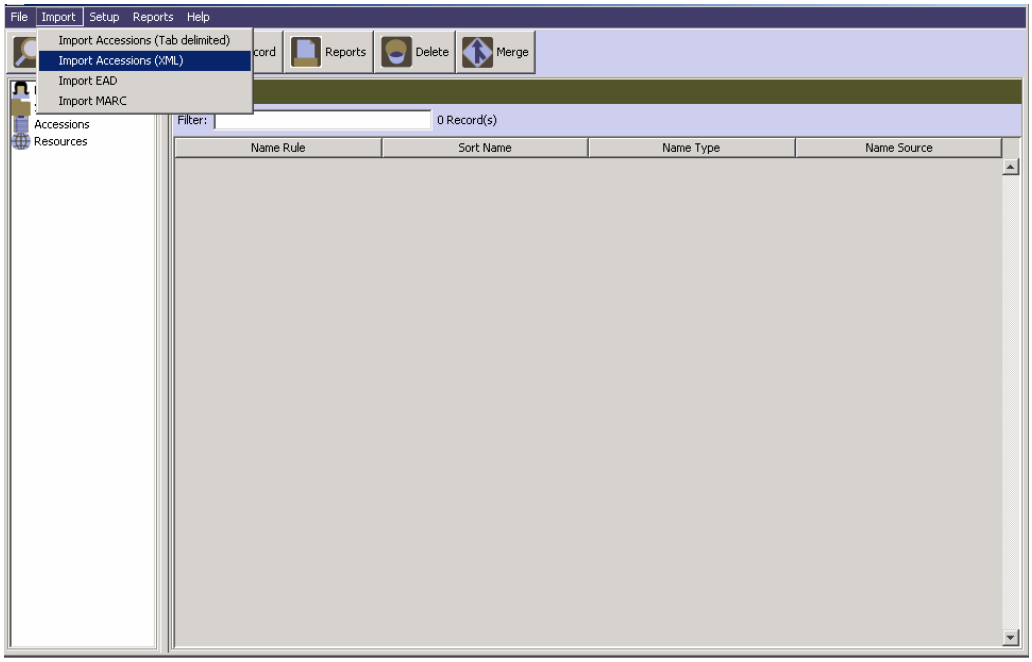
If any record does not follow the validation rules listed in the appendices, import of that record will fail.

Importing XML accessions data

XML formatted accession data can also be imported into the Toolkit. The XML format allows for multiple names and subjects types to be imported. The XML schema needed to create XML formatted data is named accessionsImport.xsd and can be found in the "conf" folder in the directory where the Toolkit is stored. Sample documents are provided in the appendices.

To import XML accessions data:

1. From the **Import** menu, select **Import Accessions (XML).**



2. Make the following selections:

    a. Choose the file to be imported.

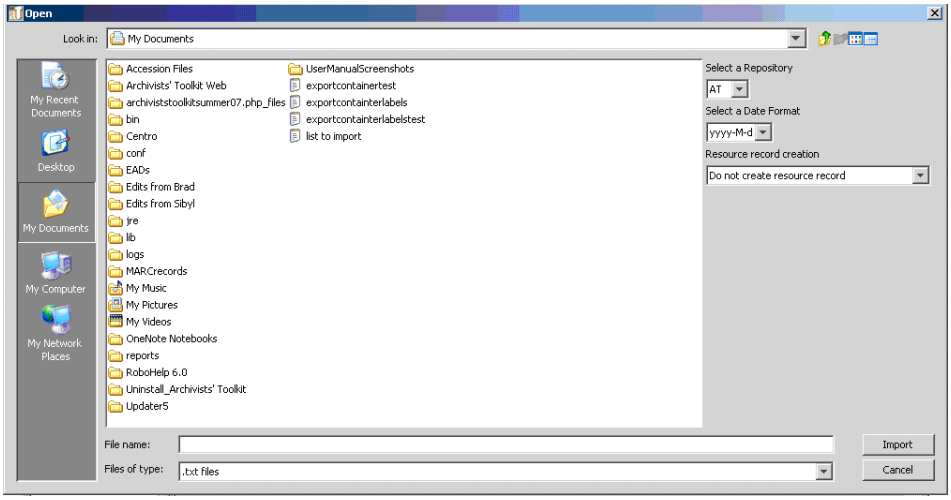
    b. Select the repository to which the accession data applies.

    c. Select the appropriate **Resource record creation** option:

    Do not create resource record. No resource records will be created; only accession records. If there is a **resource identifier** in the import file the system will check to see if the resource exists and link to it.

    Create resource with resource id only. If there is a **resource identifier** in the import file the system will check to see if the resource exists and link to it. If not, a new resource record will be created. Only the **resource identifier** and **repository** fields will be present in the resource record; all other fields will be empty.

    Create resource record using all fields. If there is a **resource identifier** in the import file the system will check to see if the resource exists and link to it. If not, a new resource record will be created. All of the fields that can be transferred from an accession record will be populated in the resource record. See Chapter 7 for a table listing how these fields are mapped.



3. Press the Import button to begin.

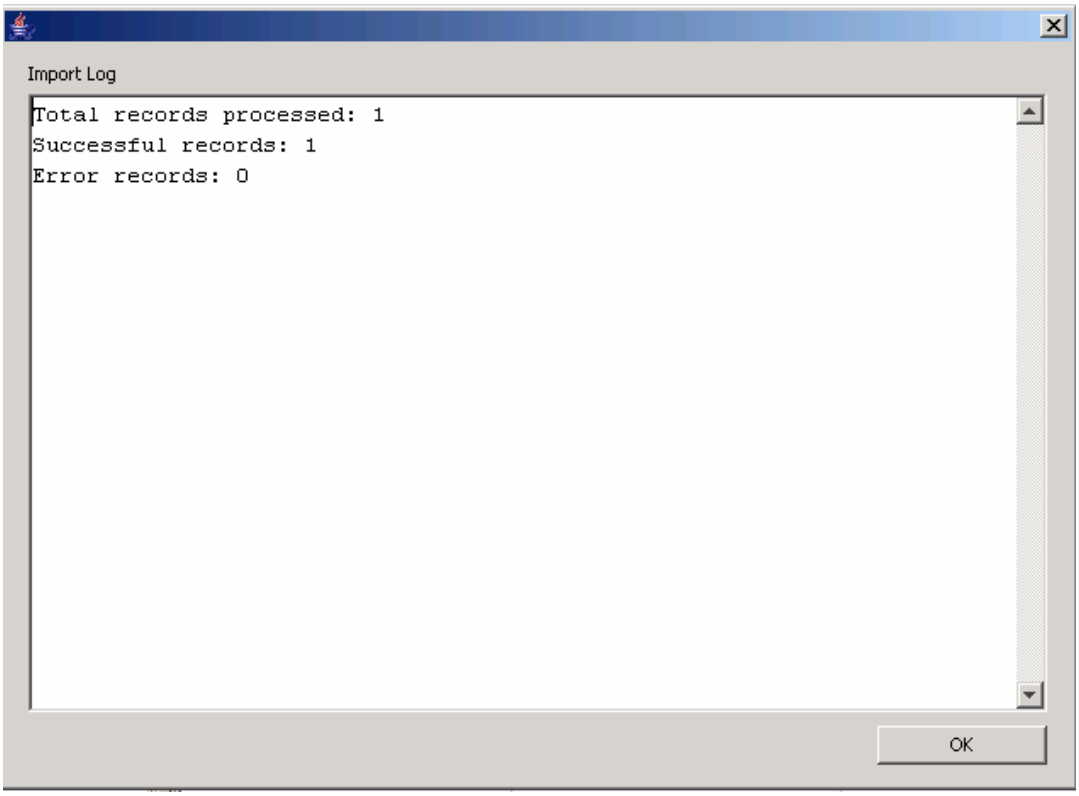
A progress window will track the import process. If errors are encountered, an error log will appear at the end of the process.

Note: Several error conditions can cause the import to fail, in whole or in part:

- An improperly formatted import document will cause the entire process to fail. No records will be imported.
- An invalid record will not be imported. To be valid an accession record must have an accession number and an accession date. A list of these validation rules is provided in the appendices.

If the import document is formatted correctly, the process will proceed record by record. All valid and error free records will be imported. All invalid and / or errant records will not be imported and will be listed as such in the resulting log.

If no errors are encountered, you will see a window like the one shown directly below.



Legacy Data Cleanup

Overview

Importing legacy data will likely require data cleanup, either before the data is imported, or within the Toolkit once the data have been imported. Failure to cleanup legacy data will lead to subsequent problems within the Tool; it is recommended that data cleanup be a top priority after import is completed.

Potential data cleanup issues

The following issues may present themselves in data that is imported into the Toolkit:

1. **Import of invalid records**. In most cases, the Toolkit will import resource records that do not contain required fields. These fields must be completed before any additional edits to the record can be saved. These invalid records can cause problems with operations such as merging of subject or name terms or merging of items in a lookup list.
2. **Import of repetitive terms in lookup lists**. When importing into fields controlled by lookup lists, the Toolkit will import data that doesn't match elements in the lookup list. For example, you may import "aat" into the **Subject Source** field, where the default value in the lookup list is "Art & Architecture Thesaurus (aat)." At times the terms may appear to be the same, but trailing whitespace causes the two strings to be different. The **Merge Item**

feature in the lookup lists can assist you with cleaning up this kind of inconsistency in your data.

3. **Import of punctuation repeated by built-in operations in the Toolkit.** In many instances, the Toolkit automatically provides punctuation for subject and name terms. If your subject and name data is imported with punctuation, there may be repetition in the Toolkit's displays and outputs.

Methods for data cleanup

Correcting invalid records

Each record in the AT has required fields. The Toolkit can import resource records that are missing these fields, and the import log will highlight which imports have these absences associated with them. These fields must be corrected before any additional information can be stored in the record. Invalid records can also create problems when merging data; since merging data is a necessary method for cleaning up other data errors, it is important to fill in the required fields for all imported records first. To correct these fields, each imported record needs to be opened, the field data entered, and then saved. The application will alert users to which required fields are necessary if they are not completed.

The Merge function

The Toolkit offers a **Merge** feature for subjects and names, as well as lookup lists, which is useful for management and clean up of your authority records and lookup lists. Merging two terms together results in the undesirable or redundant heading or list item being deleted, and all of its linked accession and description records being linked to the more desirable heading or list item. You might use this feature to perform clean up if importing data has resulted in redundant records or if you find that related terms have been used inconsistently.

Refer to Chapter 11 for instructions on merging Name and Subject headings.

Refer to Chapter 15 for instructions on merging lookup list items.

Correcting data mapping

Due to the flexibility of EAD, some instances of imported data will not map to the desired field. Examples of this are listed in the section above on Mapping of EAD elements to AT fields. Moving the data into the correct fields will ensure that records are exported correctly.