

TOMES Software User Guide

December 2018

Introduction

This document describes the TOMES software workflow and the software modules that make each step possible. Individual descriptions of each module will include links to the module's repository on [GitHub](#) where the source code and documentation can be accessed.

About TOMES

The *Transforming Online Mail with Embedded Semantics* (TOMES) project ran from 2015 to 2018. It was funded by the National Historical Publications and Records Commission.

The project aimed to identify email accounts of public officials with enduring value to capture, preserve, and provide access to government records.

The TOMES project partners included the Kansas State Historical Society, the Utah State Archives, and the State Archives of North Carolina.

The official project page can be accessed online at <https://www.ncdcr.gov/resources/records-management/tomes>.

Software Workflow and Overview

The TOMES software addresses the software deliverable portion of the TOMES project.

The goals of the software are to implement the project software workflow that consists of following steps:

1. PST conversion to EML
2. EML or MBOX conversion to EAXS (Email Account XML Schema)
3. EAXS conversion to a "tagged" EAXS file
4. Creation of an Archival Information Package (AIP) consisting of source account files, their derivatives (EAXS, etc.), and basic METS files

All TOMES software is experimental. The user assumes all risk. The use of preservation master files in conjunction with the software is not recommended. Use copies instead.

EAXS

EAXS is an XML schema designed to store XML-encoded message and attachment data for a single email account. The schema was developed by the State Archives of North Carolina and the Smithsonian Institution Archives circa 2008 for the *Electronic Mail Capture and Preservation Grant* (EMCAP).

The EAXS schema currently resides at the [tomes-eaxs](#) GitHub repository.

Tagged EAXS

A tagged EAXS file is a variation of EAXS developed specifically for the TOMES project.

A tagged EAXS file serves to store an additional, semantic version of each email message found within the source EAXS. This semantic version of the message is XML-encoded and contains markup to denote named entities found within the message by Named Entity Recognition (NER) software.

Types of named entities include persons, places, and organizations. Additionally, entities may include markings for Personally Identifiable Information (PII) and other government-centric entity patterns developed by the project members. All entities created for the project are notated in an *entity dictionary*.

More information on the semantic version of messages inside tagged EAXS files can be found in the documentation at the [tomes-tagger](#) GitHub repository.

Entity Dictionaries

Entity dictionaries are files that contain pattern matching syntax for custom named entities. These patterns are used by NER software while creating tagged EAXS files.

The entity dictionary created by the project team, *TOMES_Entity_Dictionary.xlsx*, can be found within the [tomes-project](#) GitHub repository.

Software Workflow Modules

The TOMES software workflow is implemented by individual modules developed for the project.

Each module has its own GitHub repository that contains documentation outlining the module's purpose, installation instructions, and usage examples. All modules can be executed as command line scripts or as native Python libraries.

The modules are listed below to match the workflow steps:

1. [TOMES PST Extractor](#): converts PST to EML
2. [TOMES DarcMail](#): converts EML or MBOX to EAXS
3. [TOMES Entities](#): converts Microsoft Excel files to a valid entity dictionary file
4. [TOMES Tagger](#): converts EAXS to a tagged EAXS file
5. [TOMES Packager](#): creates an AIP structure consisting of source and derivative files as well as basic METS files

Users wanting to create a custom entity dictionary must reference the documentation for the **TOMES Entities** module. It should be noted that creating advanced pattern matching syntax for an entity dictionary requires knowledge of regular expressions.

The **TOMES Packager** documentation contains information on how advanced users can customize METS output. This is recommended only for users with expertise in XML and METS as well as programming knowledge.

TOMES Software Usage

The remainder of this document covers usage of the TOMES software, specifically:

- 1) Installation and Command Line Usage of Individual Modules
- 2) Sample Account Project
 - a. Starting with a sample PST file, this sample project will use the TOMES software to create a TOMES AIP consisting of the source PST and derivate data: an EML folder, EAXS and tagged EAXS files, and METS files.
- 3) TOMES Docker
 - a. Recommended only for interested Linux users familiar with Docker and web servers, this section covers a web-interface to the TOMES software used for in-project account processing by the State Archives of North Carolina.

Installation and Command Line Usage of Individual Modules

Command line usage of TOMES software requires proficiency with the command line and PowerShell (Windows) or Bash (Mac/Linux). Prior knowledge of installing Python 3 and using the Python package manager "pip" will be extremely helpful.

This section will assume the user has Windows, therefore absolute folder paths will use a Windows-style of notation.

Create a TOMES folder

To start, create a folder in which to download ZIP files of the TOMES software modules, e.g. "C:\TOMES".

Download and Extract Module ZIP files

Download the following applications as ZIP files from the following GitHub repositories:

1. [TOMES PST Extractor](#)
2. [TOMES DarcMail](#)
3. [TOMES Entities](#)
4. [TOMES Tagger](#)
5. [TOMES Packager](#)

Extract each ZIP file directly into "C:\TOMES". For example, when you unzip **TOMES PST Extractor** and extract it to "C:\TOMES" you should have a folder called "C:\TOMES\tomes-pst-extractor-master". Inside this folder should be subfolders named "docs", "tests" and "tomes_pst_extractor" as well as some files.

Install Modules

Each module folder contains a "docs/documentation.md" file. This file can be read in a text editor or online at the corresponding GitHub repository page.

For each module, read the corresponding documentation file and install the software as indicated. Some modules will be easier to install than others given the difference in dependencies.

Run Sample Commands

Each documentation file also demonstrates how to use the software from the command line and how to view the "help" strings for each module with the "-h" flag. The help strings for each module display a sample command.

For each module, run all sample commands referenced by the help strings and view the resulting data files and folders.

Sample Account Project

After installing the TOMES software modules and developing familiarity with the sample commands, the individual modules can be used in sequence to complete a sample project.

This sample project will perform the following workflow:

Workflow Step	Workflow Module
Create a root AIP folder	N/A
Convert a sample PST file to EML	TOMES PST Extractor
Convert the EML to EAXS	TOMES DarcMail
Convert EAXS to tagged EAXS	TOMES Tagger
Create a TOMES AIP	TOMES Packager

Note that the **TOMES Entities** module will not be used. The sample project will use the entity dictionary already created for the TOMES project.

Create a root AIP folder

Create a root project folder called "sample_account":

```
PS C:\TOMES> mkdir sample_account
```

Now, create subfolders within the "sample_account" directory:

```
PS C:\TOMES> mkdir sample_account/pst
```

```
PS C:\TOMES> mkdir sample_account/mime
```

```
PS C:\TOMES> mkdir sample_account/eaxs
```

These subfolders comply with the AIP structure imposed by the **TOMES Packager** module.

Convert a sample PST file to EML

First, copy a sample PST file into the "pst" subfolder:

```
PS C:\TOMES> cp tomes-pst-extractor-master/tests/sample_files/sample.pst
sample_account/pst
```

Finally, convert the PST file to an EML:

```
PS C:\TOMES> cd tomes-pst-extractor-master/tomes_pst_extractor

PS C:\TOMES> py -3 pst_extractor.py sample_account ../../sample_account/pst/sample.pst
../../sample_account/mime
```

The resulting EML folder, "sample_account", is now located at "C:\TOMES\sample_account\mime".

Convert the EML to EAXS

Convert the EML to EAXS:

```
PS C:\TOMES\tomes-pst-extractor-master\tomes_pst_extractor> cd ../../

PS C:\TOMES> cd tomes-darcmail-master/tomes_darcmail

PS C:\TOMES\tomes-darcmail-master\tomes_darcmail> py -3 darcmail.py sample_account
../../sample_account/mime ../../sample_account
```

The resulting EAXS data is now located at "C:\TOMES\sample_account\eaxs\sample_account".

Convert EAXS to tagged EAXS

First, start the Stanford CoreNLP server by double-clicking the startup file "C:\TOMES\tomes-tagger-master\NLP\stanford_edu\start.bat" file. Per the documentation for **TOMES Tagger**, you must have already downloaded the CoreNLP software for the startup file to work.

Finally, create a tagged version of the EAXS file:

```
PS C:\TOMES\tomes-darcmail-master\tomes_darcmail> cd ../../

PS C:\TOMES> cd tomes-tagger-master/tomes_tagger

PS C:\TOMES\tomes-tagger-master\tomes_tagger> py -3 tagger.py
../../sample_account/eaxs/sample_account/eaxs_xml/sample_account.xml
../../sample_account/eaxs/sample_account/eaxs_xml/sample_account__tagged.xml
```

The resulting tagged EAXS file, "sample_account__tagged.xml", is now located in "C:\TOMES\sample_account\eaxs\sample_account\eaxs_xml".

Create a TOMES AIP

Create a TOMES AIP:

```
PS C:\TOMES\tomes-tagger-master\tomes_tagger> cd ../../

PS C:\TOMES> cd tomes-packager-master/tomes_packager
```

```
PS C:\TOMES\tomes-packager-master\tomes_packager> py -3 .\packager.py sample_account
../.. ../..
```

METS files are now located in "C:\TOMES\sample_account".

TOMES Docker

The [tomes-docker](#) GitHub repository contains an experimental Docker-based application that provides access to an earlier subset of the TOMES project repositories as a web application.

Specifically, **TOMES Docker** provides access to earlier versions of the following:

- 1) TOMES PST Extractor
- 2) TOMES DarcMail
- 3) TOMES Tagger

The web application was used for in-project account processing by the State Archives of North Carolina on remote servers. It is documented below to illustrate features of a potential, future graphical interface to the TOMES software. In any case, the aforementioned command line versions of the TOMES software are the most up-to-date and recommended versions.

TOMES Docker is not known to work with more recent versions of Docker for Windows. It is recommended only for interested Linux users familiar with Docker, web servers, and Linux file permissions. As such, Linux-style paths will be used below.

Installation

TOMES Docker requires the following dependencies:

- 1) [Docker](#)
- 2) [Docker Compose](#)

After installing the dependencies, download [tomes-docker](#) as a ZIP file. Unzip the file to the location of your choice.

Open a terminal in the "run_environments" subfolder and enter the following command:

```
$ docker-compose up
```

Open your browser to <http://localhost:80/>.

You should see the following screen:

Main
Convert a PST
Convert to EAXS
Tag EAXS
Documentation

TOMES (Transforming Online Mail with Embedded Semantics)

Welcome to the TOMES application. From here you can:

- Convert PSTs to MBOX or EML: We recommend EML at the moment
- Convert MBOXs or EMLs to EAXS (an XML representation of a Mailbox)
- Tag the EAXS with Named Entities
- Search the Tagged Mailbox

You are not obligated to do all of these steps you. The TOMES app will output a fully documented AIP at every step.

Technologies in TOMES

The TOMES app utilizes multiple technologies. Below are a listing of the main software components:

Docker	Docker is used to link all of the apps together and provide a simple setup and execution across platforms.
PST to Mime	<p>The PST to Mime part of the tool utilizes an Afterlogic's MailBee .NET Objects library to convert PSTs to Mime email. Currently, those mime emails are .EMLs.</p> <p>MailBee is the only closed source part of this project. The project has has purchased a license for the project and binaries may be distributed.</p>
DarcMail	<p>TOMES utilizes a fork of CmdDarcMailXML written by Carl Schaefer of the Smithsonian Institution.</p> <p>DarcMail converts a MBOX or an EML structure into EAXS xml.</p>
Stanford Core NLP	Messages are tagged using Stanford's Core NLP engine. TOMES provides a method for using a tagging dictionary which allows for States and organizations to customize the tags for semantics relevant to their organizational context.

Copyright 2017 by [Tomes Project](#).

The application server can be stopped at any time by entering the following command:

```
$ docker-compose down
```

Data Files

Data files (PST, EAXS, etc.) are accessible to the application in your system's `"/mnt/data"` folder.

Source PST files must be placed in the `"/mnt/data/pst"` subfolder, e.g. `"/mnt/data/pst/sample.pst"`.

Likewise, source EML or MBOX folders must be placed in the `"/mnt/data/mboxes"` subfolder.

EAXS and tagged EAXS files created by the application will be placed in the `"/mnt/data/eaxs"` subfolder.

PST to EML Conversion

To convert a PST file to EML, first place a sample PST file in `"/mnt/data/pst"`.

Now, click the web application's "Convert a PST" tab. You should see the following screen:

The screenshot shows the 'Convert a PST' tab of a web application. At the top is a navigation bar with tabs: 'Main', 'Convert a PST' (active), 'Convert to EAXS', 'Tag EAXS', 'Rectangular Snp', and 'Documentation'. Below the navigation bar, the instruction 'Select a PST to convert to mime email.' is displayed. Underneath is a folder icon with a '+' sign and the label 'PSTS'. Below that is a text input field labeled 'Output Mime-type folder:'. At the bottom is a 'Convert' button.

Click the "+" sign to the left of the yellow "PSTS" folder image to show a drop-down list of available PST files within `"/mnt/data/pst"`.

Now select the PST file to convert by clicking it. This will populate the "Output Mime-type folder" box with a recommended output name. It needed, edit this string. Use only numbers, letters, and underscores if possible.

Finally, click the "Convert" button.

The resulting EML folder will appear as a subfolder in `"/mnt/data/mboxes"`.

EML/MBOX to EAXS Conversion

To convert an EML or MBOX folder to EAXS, first place a sample EML or MBOX folder in `"/mnt/data/mboxes"`. Alternately, if you just converted a PST to EML (as above) an EML folder will already exist.

Now, click the web application's "Convert to EAXS" tab. You should see the following screen:

The screenshot shows the 'Convert to EAXS' tab of the web application. The navigation bar is the same as the previous screenshot, but the 'Convert to EAXS' tab is now active. Below the navigation bar, the instruction 'Select a directory containing Mime Emails to transform.' is displayed. Underneath is a folder icon with a '+' sign and the label 'Mime Sources'. Below that is a text input field labeled 'Transfer Name:'. Further down are three toggle switches: 'Chunk output?' (set to 1000), 'Stitch chunks?', and 'Is source .EML?'. At the bottom is a 'Convert' button.

Click the "+" sign to the left of the yellow "Mime Sources" folder image to show a drop-down list of available EML or MBOX folders within `"/mnt/data/mboxes"`.

Now select the EML or MBOX folder to convert by clicking it. This will populate the "Output Mime-type folder" box with a recommended output name.

If the source folder is an MBOX, toggle the "Is source EML?" box to the off position. The other toggle boxes should remain in their original position regardless of source type.

Finally, click the "Convert" button.

The resulting EAXS data will appear in a subfolder in "/mnt/data/eaxs".

EAXS to tagged EAXS Conversion

After creating EAXS data, click the web application's "Tag EAXS" tab. You should see the following screen:

The screenshot shows a web application interface with a navigation bar at the top containing five tabs: "Main", "Convert a PST", "Convert to EAXS", "Tag EAXS" (which is highlighted), and "Documentation". Below the navigation bar, the text "EAXS file to transform:" is followed by a folder icon and the label "EAXS". Below this, there are two input fields: "EAXS Selected" and "Tagged Name". At the bottom left of the form area is a button labeled "Convert".

Click the "+" sign to the left of the yellow "EAXS" folder image to show a drop-down list of available EAXS folders within "/mnt/data/eaxs".

Now select the EAXS to convert by clicking it. This will populate the "EAXS Selected" and "Tagged Name" boxes.

Finally, click the "Convert" button.

The resulting tagged EAXS data will appear in the original EAXS subfolder in "/mnt/data/eaxs".

APPENDIX A: Glossary

PST

The Personal Storage Table (PST) format is an open proprietary format used to store archived items and maintain offline availability. A PST file is a stand-alone, self-contained, binary file containing Folder Objects which hold Message Objects that may contain Attachment Objects. Products such as Microsoft Office and Office 365 provide the option to export entire accounts in PST format.

Source: <https://www.loc.gov/preservation/digital/formats/fdd/fdd000378.shtml>

EML

Electronic Mail Format (EML) files are saved in the Internet Message Format protocol. EMLs are compliant with the Internet Message RFC 2822 format and usually store each message in a single file.

Source: <https://www.loc.gov/preservation/digital/formats/fdd/fdd000388.shtml>

MBOX

MBOX is a family of related file formats holding all the messages of a folder in a single database file. Individual email messages are appended to the end of the file. Messages are stored in their original Internet Message RFC 2822 format. The four variations of MBOX are MBOXO, MBOXRD, MBOXCL, and MBOXCL2.

Source: <https://www.loc.gov/preservation/digital/formats/fdd/fdd000383.shtml>

NER

Named Entity Recognition (NER) is a form of Natural Language Processing that employs various disciplines such as computer science and linguistics to automatically label objects within machine-readable text. Objects may be labelled as persons, places, and organizations. NER software applications may be customized to include additional types of object labels such as Personally Identifiable Information (PII).