

# EndNote®

## ...Bibliographies Made Easy™

### Document Version 1.0

#### Introduction

This document contains examples and explanations of the AppleScript® support offered in EndNote® X1 for Apple® Macintosh® computers.

EndNote® X1 is the latest version of the industry standard software tool used by millions of researchers, writers, students, and librarians for publishing and managing bibliographies and reference lists.

AppleScript® is an English-like language developed by Apple® and used to write script files that automate the actions of the computer and its applications.

For more information from Apple®:

<http://www.apple.com/macosx/features/applescript/>

<http://developer.apple.com/AppleScript/>

#### Commands

Along with support for the AppleScript® *Standard Suite* and the *Text Suite*, the following EndNote® commands and classes are available as part of *EndNote Services Scripting*. These commands are accessible from the "Open Dictionaries" command within the Apple® *Script Editor* application available on most Macintosh® computers.



**application** [inh. NSCoreSuite.NSApplication] : EndNote application. Has been extended to include the property "Output Styles":

**output styles** (Unicode text) : The list of available output styles.

#### Example:

```
tell application "EndNote X1"  
    output styles  
end tell
```

**Result:**

List of all output styles available in the EndNote X1 application.



**document** [inh. NSCoreSuite.NSDocument] : EndNote document.

The following commands were added:

**import** : Import records into a document.

**import** specifier : the document into which records are to be imported

**records from** string : RXML data as unicode string, to be imported into the document

**Example:**

```
tell application "EndNote X1"
  set res to find "dinosaur"
  set exportedRecs to export records res as "RXML"
  make new document at beginning with properties {name:"dinosaurDB"}
  set doc to first document
  import doc records from exportedRecs
end tell
```

**Result:**

New Database called "dinosaurDB.enl" is created and populated with the records containing the word "dinosaur" from Paleo.enl.

**See Also:** **export records** command.



**delete record** : Delete the record with specified ID(s)

**delete record** list of string : single Record ID, or list of record IDs

**Example:**

```
tell application "EndNote X1"
  set res to find "dinosaur"
  delete record first item of res
end tell
```

**Result:**

Record #1 is deleted from Paleo.enl.



**export** : Export records from a document.

**export** specifier : the document from which records are to be exported.

**Example:**

```
tell application "EndNote X1"
    set res to find "dinosaur"
    set exportedRecs to export records res as "RSXML"
    make new document at beginning with properties {name:"dinosaurDB"}
    set doc to first document
    import doc records from exportedRecs
    set exportedRecs2 to export doc
end tell
tell application "Finder"
    set f to "test.xml"
    set ref_num to open for access f with write permission
    set eof of ref_num to 0
    write exportedRecs2 to ref_num
    close access ref_num
end tell
```

**Result:**

The entire contents of dinosaurDB.enl are exported as XML to Test.xml.

**Note:** The 'export records' command exists to exports records based on a list of records. The plain 'export' command described above is basically a shortcut to export **all** the records from a given database. For instance, one could set myXML to export first document and, assuming the first document is a library, the result would be XML for all that library's records. Using the 'export records' command, one would need to create a record list containing all IDs to achieve the same effect.



**export records** : Export a list of records with specified format  
**export records** list of string : single Record ID, or list of record IDs  
**as** string : Specifier of format to export. (Currently RSXML is the only supported format).

**Example:**

```
tell application "EndNote X1"
    set res to find "dinosaur"
    set exportedRecs to export records res as "RSXML"
end tell
tell application "Finder"
    set f to "test.xml"
    set ref_num to open for access f with write permission
    set eof of ref_num to 0
    write exportedRecs to ref_num
    close access ref_num
end tell
```

**Result:**

Similar to exporting records as XML, saving to file /test.xml.

**See Also:** **export** command.



**fields of** : Get a list of names of fields of a record.

**fields of** string : The ID of the record containing the desired fields

**Example:**

```
tell application "EndNote X1"
    set res to find "dinosaur"
    fields of first item of res
end tell
```

**Result:**

```
{"Author", "Year", "Title", "Series Editor", "Series Title", "City", "Publisher", "Volume", "Extent of Work", "Number", "Edition", "Date", "Type", "Performers", "Short Title", "Alternate Title", "ISBN", "DOI", "Contents", "Cast", "Credits", "Size/Length", "Format", "Accession Number", "Call Number", "Label", "Keywords", "Abstract", "Notes", "Research Notes", "URL", "File Attachments", "Author Address", "Figure", "Caption", "Access Date", "Last Modified Date", "Translated Author", "Translated Title", "Name of Database", "Database Provider", "Language"}
```



**find** : find records containing text

**find** string : text to find in records of open libraries

[**in field** string] : optional: name of field to search in each record (if omitted all fields are searched)

**Example:**

```
tell application "EndNote X1"
    find "dinosaur"
end tell
```

**Or:**

```
tell application "EndNote X1"
    find "dinosaur" in field "title"
end tell
```

**Result:**

**List of Record IDs**

```
{"<RecordID database=\"1\" recNum=\"1\" />", "<RecordID database=\"1\" recNum=\"3\" />", "<RecordID database=\"1\" recNum=\"4\" />", "<RecordID database=\"1\" recNum=\"6\" />", "<RecordID database=\"1\" recNum=\"11\" />", "<RecordID database=\"1\" recNum=\"22\" />", "<RecordID database=\"1\" recNum=\"48\" />", "<RecordID database=\"1\" recNum=\"52\" />", "<RecordID database=\"1\" recNum=\"125\" />", "<RecordID database=\"1\" recNum=\"137\" />", "<RecordID database=\"1\" recNum=\"162\" />", "<RecordID database=\"1\" recNum=\"180\" />", "<RecordID database=\"1\" recNum=\"185\" />", "<RecordID database=\"1\" recNum=\"186\" />", "<RecordID database=\"1\" recNum=\"195\" />", "<RecordID database=\"1\" recNum=\"211\" />"}
```



**format manuscript** : Takes a manuscript spec and generates a formatted bibliography in RTF  
**format manuscript** string : manuscript specification in XML format

**Example:**

```

tell application "EndNote X1 Debug"
  set recs to find "1984"
  set rec to second item in recs
  set res1 to get record contents rec
  set sub to text 49 thru ((length of res1) - 10) of res1
  set param to "<format-manuscript><output-style>" & "Show All Fields" & "</output-
style><starting-refnum>1</starting-refnum><citations><citation exclude-auth=\"true\" exclude-
year=\"true\"><contiguous>true</contiguous><note-id>0</note-
id><prefix>&#xA; </prefix><suffix>&#xA; </suffix>" & sub & "</citation></citations></format-
manuscript>"
  set res4 to format manuscript param

```

**end tell**

**Result:**

Record formatted as Rich Text Format, using the "Show All Fields" Bibliographic Output style.

```

"{\rtf1\ansi\ansicpg1252\deff1\deflang1033{\fonttbl{\f1\fnil\fs24\prq0
Arial;}}\margl1440\margr1440\margt1440\margb1440\sectd\titledpg\pard\li0\ri0\fi0\fs24
{\f1\fs24\b Reference Type: }{\f1\fs24 Journal Article\par }
{\f1\fs24\b Record Number: }{\f1\fs24 52\par }
{\f1\fs24\b Author: }{\f1\fs24 B. S. Bowstein\par }
{\f1\fs24\b Year: }{\f1\fs24 1984\par }
{\f1\fs24\b Title: }{\f1\fs24 Micro-mammal prototypes in dinosaurs: a new discovery\par }
{\f1\fs24\b Journal: }{\f1\fs24 Paleontology\par }
{\f1\fs24\b Volume: }{\f1\fs24 12\par }
{\f1\fs24\b Pages: }{\f1\fs24 74-78\par }
{\f1\fs24\b Short Title: }{\f1\fs24 Micro-mammal prototypes in dinosaurs: a new
discovery\par }
{\f1\fs24\b Keywords: }{\f1\fs24 Dinosaurs\par }
{\f1\fs24 \par }
{\f1\fs24 \par }"}

```



**format bibliography** : Create a formatted bibliography with a list of records

**format bibliography** string : List of records to format

[to file] : optional: file path (if omitted, results will be placed on pasteboard and can be pasted in other applications)

[with string] : optional: output style (if omitted, default style chosen in EndNote application will be used instead)

**Example:**

```

tell application "EndNote X1"
    set res to find "dinosaur"
    format bibliography res
end tell

```

**Or:**

```

tell application "EndNote X1"
    set res to find "dinosaur"
    format bibliography res with "Show All Fields"
end tell

```

**Or:**

```

tell application "EndNote X1"
    set res to find "dinosaur"
    format bibliography res to "test.rtf" with "Show All Fields"
end tell

```

**Result:**

Puts a list of formatted records on pasteboard (Example 1 above), formats records with Show All Fields output style and puts them on pasteboard (Example 2 above) Saves formatted record list to file /test.rtf (Example 3 above).



**get record contents** : Get the content of the record formatted as XML  
**get record contents** list of string : single Record ID, or list of record IDs

**Example:**

```

tell application "EndNote X1"
    set res to find "dinosaur"
    get record contents first item of res
end tell

```

**Result:****XML formatted record**

```

"<?xml version=\"1.0\" encoding=\"UTF-8\" ?><records><record><database name=\"Paleo.enl\"
path=\"/Applications/EndNote X1/Examples/Paleo.enl\">Paleo.enl</database><source-app
name=\"EndNote\" version=\"10.0\">EndNote</source-app><rec-number>1</rec-
number><foreign-keys><key app=\"EN\" db-
id=\"2a90f9exlrssese5pzgppfyer0s9as5w5v5\">1</key></foreign-keys><ref-type
name=\"Artwork\">2</ref-type><contributors><authors><author><style face=\"normal\"
font=\"default\" size=\"100%\">Fortescue,
Julian</style></author></authors></contributors><titles><title><style face=\"normal\"
font=\"default\" size=\"100%\">The enigma of the extinction of the
dinosaurs</style></title></titles><dates><year><style face=\"normal\" font=\"default\"
size=\"100%\">1979</style></year></dates><pub-location><style face=\"normal\"
font=\"default\" size=\"100%\">Berkeley, Calif.</style></pub-location><notes><style
face=\"normal\" font=\"default\" size=\"100%\">Constructed of plastic toy dinosaurs, cardboard

```

shoe box, tar.</style></notes><work-type><style face=\"normal\" font=\"default\" size=\"100%\">mixed media</style></work-type><urls></urls></record></records>"

**See Also:** write record command.



**get formatted record** : Get the record formatted with output style  
**get formatted record** list of string : single Record ID, or list of record IDs

**Example:**

```
tell application "EndNote X1"  
    set res to find "dinosaur"  
    get formatted record first item of res  
end tell
```

**Result:**

Fortescue, J. The enigma of the extinction of the dinosaurs. Berkeley, Calif. 1979.  
Formatted as RTF (Rich Text Format).

```
"{\rtf1\ansi\ansicpg1252\deff1\deflang1033{\fonttbl{\f1\fnil\charset0\prq0  
Arial;}}\margl1440\margr1440\margt1440\margb1440\sectd\tilepg
```

```
\pard\li0\ri0\fi0\fs24
```

```
{\f1\fs24 Fortescue, J. The enigma of the extinction of the dinosaurs. Berkeley,  
Calif. 1979.}\par }
```

```
{\f1\fs24 \par }}"
```



**get unformatted record** : Get the unformatted citation of the record  
**get unformatted record** list of string : single Record ID, or list of record IDs

**Example:**

```
tell application "EndNote X1"  
    set res to find "dinosaur"  
    get unformatted record first item of res  
end tell
```

**Result:**

```
"{Fortescue, 1979 #1}"
```



**highlight** : Highlight records in a document window.

**highlight** specifier : the document in which records are to be highlighted.

**records in** string : Record ID or list of record IDs to select in document's window

**Example:**

```
tell application "EndNote X1"
    set res to find "dinosaur"
    highlight first document records in res
end tell
```

**Result:**

Records containing the word "dinosaur" are selected in first document.



**read field** : Read the field of a record

**read field** string : field name to read

**of record** string : Record ID to read

**Example:**

```
tell application "EndNote X1"
    set res to find "dinosaur"
    read field "Author" of record first item of res
end tell
```

**Result:**

"Haines, Tim"

**Note:** Field name is case sensitive and matches the name in the reference type of the target record.

**See Also:** write text command.



**show** : Show only the specified list of records in the library window.

**show** specifier : the document in which records are to be selected.

**records in** string : Record ID or list of record IDs to display in document's window

**Example:**

```
tell application "EndNote X1"
    set res to find "dinosaur"
    show first document records in res
end tell
```

## end tell

### Result:

Only records containing the word "dinosaur" are displayed in the first document.



**write** : Modify or create one record in a document window.  
**write** specifier : the document in which records are to be written.  
[**record** string] : Record ID to modify in document's window  
**content** string : Record contents as XML

### Example:

```
tell application "EndNote X1"  
    set res to find "dinosaur"  
    set exportedRecs to export records res as "RSXML"  
    make new document at beginning with properties {name:"testDB"}  
    set doc to first document  
    set rec to get record contents first item of res  
    write doc content rec  
end tell
```

### Result:

A new database: testDB is created with a copy of 1 record from Paleo.enl.



**write record** : Update the record or create new record with specified contents  
**write record** string : contents of record formatted as XML  
**with ID** string : Record ID as unicode string

### Example:

```
tell application "EndNote X1"  
    set res to find "dinosaur"  
    set recStr to get record contents first item of res  
    set newRecStr to text 1 thru 706 of recStr & "1983" & text 711 thru last character of  
recStr  
    write record newRecStr with ID first item of res  
end tell
```

### Result:

Most AppleScript commands do not refresh display, but opening Record #1 ({{Fortescue, 1979 #1}}) will confirm that the year has been changed to 1983:

Fortescue, J. The enigma of the extinction of the dinosaurs. Berkeley, Calif. 1983.

**Note:** This is a fairly difficult way of changing records, more suited to making large scale changes.

A more efficient way to update fields is using write <text> to field <field name> of record <record ID>, discussed above.



**write text** : Write the field of a record  
**write text** string : fields contents to write  
**to field** string : fields name to write to  
**of record** string : Record ID to read

**Example:**

```
tell application "EndNote X1"  
    set res to find "dinosaur"  
    write text "James, Jesse" to field "Author" of record first item of res
```

**Result:**

Most AppleScript commands do not refresh display, but opening Record #3 shows that "Haines, Tim" changed to "James, Jesse".

**Note:** Field name is case sensitive and matches the name in the reference type of the target record.

**Sample Scripts**

The sample scripts use the commands defined in the previous section and help to demonstrate some common uses of the EndNote® AppleScript® functions, including:

1. Opening an existing database (EndNote® Library).
2. Doing a 'Keyword' search.
3. Creating a formatted bibliography from the results.
4. Exporting the results to RSXML.
5. Creating (make) a new database.
6. Importing RSXML into that database.
7. Reading data from the 'Author' field of records.
8. Selecting (highlight) a set of records.
9. Showing a set of records.
10. Changing the 'Notes' field in a set of records.
11. Closing the database opened and created.

Actual script samples can be found in the accompanying file "X1 AppleScript Sample.scpt" available from the EndNote® website – <http://www.endnote.com>.

The EndNote® RSXML DTD may be helpful and is also available from the EndNote® website – <http://www.endnote.com>.

The sample scripts use the Paleo.enl sample library installed with EndNote® X1.

```

tell application "EndNote X1"
  -- Let's open Paleo from the Examples folder.
  open (path to applications folder as string) & name & ":Examples:Paleo.enl"

  -- open doesn't return anything, but we need to remember it for close later on
  set myPaleo to front document

  -- And see what we can find.
  set myResults to find "dinosaur" in field "keywords"

  -- Let's create a reading list of the results of the 'dinosaur' keywords search in an RTF
  file
  -- Note: 'format bibliography' requires a POSIX path (meaning slashes, not colons)
  set myBibFile to POSIX path of (path to desktop as alias) & "DinoReading.rtf"
  format bibliography myResults to myBibFile with "Annotated"

  -- It would be nice to export this as XML for another script to consume
  set myExport to export records myResults as "RSXML"
end tell

  -- Using AppleScript® commands, we write the xml out to a file.
  -- Note: The AppleScript® extension command 'write' conflicts with EndNote's
  -- command of the same name. To use AppleScript® 'write', place it outside of
  -- EndNote's 'tell' block soEndNote® will ignore it.
  set myXMLFile to open for access (path to desktop as string) & "DinoReading.xml"
  with write permission
  write myExport to myXMLFile
  close access myXMLFile

tell application "EndNote X1"
  -- Let's create a new database. Note that 'make' requires a POSIX path, too.
  set myDB to make new document with properties {path:POSIX path of (path to
  desktop as string) & "DinoReading.enl"}

  -- We want to import the exported records into the new database. A different script
  -- would need to read the DinoReading.xml file in, but the XML is already in
  -- 'myExport', so we'll just use that.
  set myImportList to import myDB records from myExport

  -- Let's check each record's author field for Billoski
  set IDList to {}
  repeat with recID in myImportList
    read field "Author" of record recID
    if the result contains "Billoski" then
      copy recID to end of IDList
    end if
  end repeat

```

```

-- Let's select the Billoski records
highlight myDB records in IDList
display dialog "Notice records containing Billoski as a author are selected." buttons
    {"OK"} default button 1

-- Let's show only these records
show myDB records in IDList

-- Let's make an annotation in these records about that
repeat with recID in IDList
    read field "Notes" of record recID
    write text result & return & "Billoski was involved." of record recID to field
        "Notes"
end repeat

-- Remember to close the database if you open/make it from the script. If you don't, -
-- you might get complaints that the database is in use when you try to delete it in the
-- Finder™, even if you manually close it after running the script (of course, quitting
-- EndNote® always will close it).
close myPaleo
close myDB
end tell

```