# Table of Contents

1. **INTRODUCTION** .................................................................................................................3
   1. 1 **DEFINITIONS** ..................................................................................................................3
   1. 2 **ORGANIZATION OF THIS CLASS** ..................................................................................4

2. **EXPLORING MACROS** ........................................................................................................4
   2.1 **RUNNING MACROS** ........................................................................................................4
   2.2 **WHAT DOES A MACRO DO?** ........................................................................................5
   2.3 **CREATING A SIMPLE MACRO** .......................................................................................7

3. **THE HUL MACRO EXPRESS iSITE** ....................................................................................14
   3.1 **ACCESSING THE HUL MACRO EXPRESS iSITE** ............................................................14
   3.2 **HUL MACRO EXPRESS iSITE STRUCTURE** ...................................................................15

4. **REVIEWING THE LIST OF CURRENT SHARED MACROS** ................................................15
   4.1 **ACCESSING THE SHARED MACROS LIST** .................................................................15
   4.2 **IDENTIFYING A MACRO POTENTIALLY USEFUL TO LOCAL WORKFLOW** ...............16
   4.3 **DOWNLOADING, VIEWING, AND READING SHARED MACRO FILE** .................17
       4.3.1 **Downloading the HUL Shared Macro File** ..........................................................17
       4.3.2 **Opening the Macro File** ......................................................................................18
       4.3.3 **Understanding the Macro Explorer Main Window** ...........................................19
   4.4 **UNDERSTANDING INDIVIDUAL MACRO FILE STRUCTURE** ...................................20
       4.4.1 **Opening a Macro** ..................................................................................................20
       4.4.2 **Understanding the Script Tab** ..............................................................................21
       4.4.3 **Understanding the Properties Tab** .....................................................................21
       4.4.4 **Understanding the Scope Tab** ...............................................................................22
       4.4.5 **Understanding the Security Tab** .........................................................................24
       4.4.6 **Understanding the Notes Tab** .............................................................................25
       4.4.7 **Closing the Notes Tab** .........................................................................................25

5. **IMPORTING MACROS** ........................................................................................................27
   5.1 **CREATING A LOCAL MACRO FILE** ..............................................................................27
   5.2 **IMPORTING MACROS FROM SHARED MACRO FILE TO LOCAL MACRO FILE** ........29

6. **MACRO ACTIVATION METHODS** .......................................................................................37
   6.1 **HOT KEY ACTIVATION** ................................................................................................37
   6.2 **SCHEDULE** .....................................................................................................................39
   6.3 **SHORTKEY ACTIVATION** .............................................................................................39
   6.4 **POPUP MENU** ................................................................................................................41
   6.5 **FLOATING MENU** ..........................................................................................................47
       6.5.1 **Creating a Floating Menu** ....................................................................................47
       6.5.2 **Changing a Popup Menu to a Floating Menu** ....................................................49
   6.6 **ASSIGNING AN ICON TO A MACRO** ............................................................................54

Revised March 10, 2009
© 2008 President and Fellows of Harvard College
http://hul.harvard.edu/ois/systems/aleph/docs/train_mixa.pdf
1. Introduction

The goal of this class is to provide library staff with an introduction to Macro Express and the shared macros currently available to the Harvard library community. Once you complete the class you will be able to navigate the program; download, install, and customize shared macros; and assign macro activation methods. We will also introduce you to the HUL Macro Express iSite and the resources available there: macro files, policies, documentation, and supplemental materials.

1.1 Definitions

Macro: an end-user-created program that allows the reproduction of keystrokes, mouse-clicks, and other computer functions within a single application or across multiple applications. Macros can be used to automate repetitive or common keyboard tasks, benefiting workflow, ergonomics, productivity, and training.

Macro Express: a desktop utility used to create and play back macros. Macro Express has primarily been used in HUL in conjunction with Aleph, but the program can be used to create macros for a wide range of Windows applications. The Macro Express software has two parts: an editor and a player (the installation package distributed by OIS includes both the player and the editor).

MEXpert (Macro Express Expert): a staff member, appointed by the manager of the local unit, who is responsible for evaluating, importing, localizing, distributing, and providing primary support for macros within their unit. MEXperts are also expected to familiarize themselves with and interpret policies and procedures found at the HUL Macro Express iSite. Only MEXperts have the security permissions to download the shared macro set from the iSite.

Macro File: a file (*.mex) that contains one or more macros. In Macro Express this type of file is also referred to as a macro library. (Other types of macro files, such as playable macros with the *.mxe extension, are not covered in this class.)

Helper File: an additional file that a macro reads and uses during playback. Some macros in the shared macro set utilize helper files. There are different types of helper files used by macros in the shared macro set. A text file (*.txt, *.ini, *.csv) may be used for storing data values that vary from unit to unit (or user to user), but are used consistently in the local workflow. Examples include order unit designations, staff members’ initials, or frequently-used notes. Some macros in the shared file create and delete text files for temporary processing of data.

Subroutine Macro (Submacro): a macro that is invoked by another macro. Some macros in the shared macro set utilize submacros. Some submacros may also be used as standalone macros; others are only useful in the context of the macro that calls them.

Internal Macro: a macro that is created and used in a single HUL unit. It must conform to HUL Macro Implementation and Scope Guidelines, but is not required to follow HUL macro standard naming and documentation conventions (see HUL Macro Naming and Documentation Conventions document). Internal macros may be very specific to the workflow of a particular unit or project.

Shared Macro: a macro that has been reviewed by the Aleph Macro Working Group (AWMG) and made available for download from the HUL Macro Express iSite. Shared macros must conform to HUL guidelines and follow standard naming and documentation conventions (see the HUL Macro Naming and Documentation Conventions document). Shared macros are stripped of

Revised March 10, 2009
key assignments before they are distributed: this allows local units to attach keys, popup menus, or floating menus according to local preferences and policies.

**Standard Macro:** a shared macro that has common use throughout all libraries without requiring modification. The functions performed by these macros do not vary from unit to unit. Examples of standard macros are all the macros developed for keying Unicode characters in Aleph.

**Localized Macro:** a shared macro that requires local customization before it can be implemented effectively in different units. Some customizations might be simple (e.g., editing a file to replace a string or a code), others may involve extensive editing to adapt to different workflows.

### 1.2 Organization of this Class

Upon successful completion of this class, users will be able to:

1. Log on to the HUL Macro Express iSite
2. Review existing shared macros and determine if they fit local needs
3. Download a shared macro file, read and interpret documentation of shared macros (including the need for subroutine macros and helper files)
4. Create a new macro file, import shared macros, and create categories for macro storage
5. Assign activation to macros via hot keys, ShortKeys, popup menus, floating menus, and scheduling
6. Set a location for a new macro file and helper files.

### 2. Exploring Macros

#### 2.1 Running macros

The Macro Express software consists of two parts: the Editor, which is used for writing macros, and the Player, which is used for running macros. (There are some other ways of using Macro Express, in addition to these, but those are beyond the scope of this class.)

The Macro Express Player must be running and the Macro Express file (*.mex) that contains the macros you want to run must be active in order for macros to play back. The Macro Express icon appears in the system tray when the Macro Express player is open; the player does not open any windows. If the player is not open, you can open the program from the Start menu or double-click any Macro Express file (*.mex). By default, the editor does not open when the player is launched. To open the editor, click on the Macro Express icon in the system tray, or press Ctrl-Alt-Shift-W. The editor opens a window, which can be closed without terminating the player.

When a macro is running, the Macro Express icon changes to the running man icon. To terminate a macro before it has run to the end, right-click on the running man icon or press the Scroll Lock and Pause keys together. Please note: only one Macro Express file may be open at any given time, and only one macro may be played at a time.
2.2 What does a macro do?

The purpose of a macro is to make tasks executed in various applications easier, more efficient and/or more accurate. Some macros perform functions as simple as writing out a text string; others may be complex requiring user input and considerable programming skill on the part of the macro developer.

Example of a simple macro script: This simple, two-line macro types out a line of text: “Sent to HD – [today’s date].”
**Example of a complex macro:** This macro updates a Widener holdings record with a call number, H08 field, and XPR field with the current date, and edits the 008 and Leader, if necessary. It also adds holdings fields for accompanying material listed in 300 $$e, and 863 fields for multi-volume works entered as such in the 300 $$a (user supplies captions).

```
Macro Script

// This macro updates holdings record with 052 information, ...
// ... H08, XPR, and holdings for accompanying material listed in 300 $$e and 853/863 for
// multivolume works
// T1=Sublibrary code
Variable Set String %T1% "0"
// T2=Collection code
Variable Set String %T2% ""
// T3=Call number in Holdings record
Variable Set String %T3% ""
// T4=Classification part of call number from prompt
Variable Set String %T4% ""
// T5=Call number cutter from prompt
Variable Set String %T5% ""
// T6=WorkingRecord.txt temp field variable
Variable Set String %T6% ""
// T7=WorkingRecord.txt temp tag variable
Variable Set String %T7% ""
// T8=Call number from 050 field
Variable Set String %T8% ""
// T9=Call number from 090 field
Variable Set String %T9% ""
// T10= record format
Variable Set String %T10% ""
// T11=Temporary storage for Collection Code
Variable Set String %T11% ""
// T12 = date from 260 field
Variable Set String %T12% ""
// T13=300 subfield a
Variable Set String %T13% ""
// T14=No. of v. in multi
Variable Set String %T14% ""
// T15=Holdings leader
Variable Set String %T15% ""
```

Lists the current line and total number of lines in the macro script. This script has 1,250 lines.

Revised March 10, 2009
2.3 Creating a simple macro

Although the purpose of this class is not to teach the creation of macros (that will be covered in MEXB), a brief overview of macro function and structure will make it easier to work with shared macros.

Many macros simply replicate the keyboard actions that a user would perform in the course of work. In this exercise, we will create a simple macro that opens up Notepad and types a sentence of text.

1. To begin, go to the Start menu and launch Macro Express (Start > All Programs > Macro Express > MacroExpress 3). This will cause a Macro Express icon to display in the task bar at the bottom right of your screen. Depending on the setup of Macro Express on your computer, the Macro Explorer window may also display on screen; if it does not, left-click on the icon which will open the Macro Explorer window.
2. On the top menu, select Macros > Add Macro.

2. The Add Macro dialog box will appear. By default, Macro Express assumes the HotKey activation method. You can assign hot keys by either scrolling through the list of keys or clicking the desired combination. After you have selected a hot key (Win+G in this example), click the Scripting Editor Button.
4. We will construct this macro using the Scripting Editor. The macro will perform two actions: open Microsoft Notepad, and type a sentence of text.

Look down the alphabetical Commands list and click Windows/Programs. Select Program Launch and double-click. A dialog box will open prompting you to select the program you wish to launch. Once you have located Notepad, click OK.

*Note:* The Notepad program executable is located in C:\WINDOWS\NOTEPAD.exe. You can type in the string or locate the program by navigating to the folder.

![Diagram of Program Launch in Scripting Editor]

Double-click on the ‘Program Launch’ command in the Commands area.
Launch Program dialog box

- Type in program path and file name ...
- or click here to locate program

Program Launch command added to the Macro Script

Program Launch command added to Macro Script
Note: If you make a mistake and insert the wrong command into the Macro Script, you can delete it: select it and click on the Delete icon in the toolbar or press the Delete key on your keyboard. The command will disappear from the script. Please note that there is no Undo function in the Scripting Editor.

5. Now we will program the macro to type out a sentence. Select Text under Commands to view text command options. Then, double-click on Text Type command to open dialog box.

Select Text Type command
6. When the Text Type dialog pops up, type in a sentence. (For this exercise, type in “Congratulations! You just programmed your first Macro.”) When you are finished, click OK.

   Note: You can cut and paste text from Word or elsewhere into the Text Type box, but it will lose any formatting that it has.

   ![Text Type dialog box]

7. You have completed your first macro. We will not save the macro but will test how it runs by clicking on the Test Run button in the icon bar at the top of the macro. If you saved the macro, you can test the macro by pressing the hot key combination you selected at the beginning.

   ![Text Type dialog box]
6. Close the Macro Editor. You will see a dialog box asking you if you wish to save this macro — click on No.

**Note:** Now that you have launched Macro Express, it remains open in the background, even when the Macro Editor window is closed; the icon in the System Tray in the bottom right task bar is an indication that the Player is running. If you wish Macro Express to launch automatically when your machine turns on, you can place a shortcut in your Windows Startup folder. It is recommended that you not use the Options>Preferences> Startup option to Run on Windows Startup; this will place a shortcut to the All Users folders, which many users cannot edit on their own computer.

Revised March 10, 2009
3. The HUL Macro Express iSite

3.1 Accessing the HUL Macro Express iSite

1. Open a web browser. (In this class we use MS Internet Explorer so that we’ll all get the same dialog boxes, but you can use the browser of your choice.)

2. Type in the following URL: http://isites.harvard.edu/k1846, or from the Aleph Documentation Center (http://hul.harvard.edu/ois/systems/aleph/docs.html), choose the Client/Print/Macro Express tab and click on the link for the HUL Macro Express iSite.

3. You will be prompted for your HUID and PIN.

4. Upon successful validation of your PIN, you will be brought to the HUL Macro Express iSite main page.

*Note:* All library staff PIN numbers are valid to view the HUL Macro Express iSite. Only MEXPerts will have PIN access to the macro download portion of the iSite.
3.2 HUL Macro Express iSite Structure

- **Home Tab**: contains the welcome message, site search, and current announcements.
- **Documentation Tab**: contains documents relating to policy, references for macro programming, and tips.
- **Macro Downloads Tab**: contains macros and helper files for download, and related documentation. If you are not a MEXpert, you will see not see the macro files; only MEXperts can download macros.
- **MEXperts Tab**: contains explanations of MEXpert responsibilities and the process of becoming a MEXpert, and a list of MEXperts and their units.
- **AMWG Tab**: contains contact information for the HUL Aleph Macro Working group, and the AMWG minutes.

4. Reviewing the List of Current Shared Macros

4.1 Accessing the Shared Macros List

A spreadsheet listing the available HUL shared macros is available for download on the HUL Macro Express iSite. To access this list, click on the Macro Downloads tab of the iSite.

From either the Documentation or Macro Downloads tab, click on the List of HUL Shared Macros (xls) link. You may open or save the Excel spreadsheet to your computer.

For the purposes of our exercise, you will find a printed copy of the HUL Shared Macros list along with your training packet.

When you have finished reviewing or downloading this spreadsheet, you may need to click on the Back button in your browser. (If the spreadsheet opens in the same browser window and if you simply close it, you will leave the iSite and need to log in again.)

*Note: The List of HUL Shared Macros is also available on the Documentation tab of the iSite.*
Since this list is updated on an ongoing basis, it is best to use the Web site version, rather than a downloaded version, when doing macro selection. The first tab of the spreadsheet consists of macros developed by HUL units to work in conjunction with Aleph; Unicode diacritic macros can be found on the second tab. As shared macros are added for other Windows applications, additional tabs will be added to the spreadsheet.

See your training packet for a printed version of the list of Aleph-related macros.

The list consists of 3 columns: Macro Nickname, Contributing Unit, and Purpose.

For an explanation of macro naming conventions, see Appendix B.

4.2 Identifying a macro potentially useful to local workflow

Before making a decision to create an in-house macro, MEXperts should carefully review the current shared macro list to see if an existing macro might serve the desired purpose, either as-is, or after localization.

The HUL Shared Macros summary spreadsheet provides only brief descriptions of available macros. For detailed information about any macro, consult the documentation in the macro’s Notes tab (see Section 4.4.6), which can be found in the shared macro file.

**Exercise 1: Reviewing HUL Shared Macros Summary Spreadsheet**

Review the HUL Shared Macros Summary Spreadsheet and determine whether it lists any macro or macros to create authority records.
4.3 Downloading, Viewing, and Reading Shared Macro File

4.3.1 Downloading the HUL Shared Macro File

To download the HUL Shared Macro file:

1. Click on the Macro Downloads tab of the HUL Macro Express iSite. This will bring you to the following page:

Download page for HUL Shared Macros

2. Click on the HUL Shared Macros – Complete File (mex) link.
3. A dialog box will appear asking if you wish to open or save the file.

![File Download dialog box](image)

Downloading Macro Express file using Internet Explorer

4. Save to local or network drive – for the purposes of this class, save it to D:\User Files\.

5. Repeat steps 2-4 for the HelperFiles zip file.

**Note:** If you are a MEXpert and are not able to see the macro/helper files on the Macro Downloads page, contact Rei Diaz at OIS, x53724, to adjust your authorization level. If you are not a MEXpert and believe you should be, contact your supervisor.

### 4.3.2 Opening the Macro File

1. Go to the folder where you saved the HUL_Shared_Macros.mex file (in this class, D:\User Files) to open it.

   Double-clicking on a macro file will open it; if Macro Express is not already launched, double-clicking on the file will launch the program as well.

   When you double-click on the macro file, if another macro file is open, it will close, and this file will replace it as the active macro file. (In this way, Macro Express differs from other applications, such as Microsoft Word, in which multiple documents may be open at the same time.)

2. Depending on the Macro Express settings, the Editor window may or may not open automatically when Macro Express is launched. To open the editor window manually, left-click on the Macro Express icon in the system tray.
4.3.3 Understanding the Macro Explorer Main Window

The Macro Explorer window consists of four parts:

- **Menu Toolbar** at top, which is used to invoke a variety of functions.

- **Actions Toolbar** at left, mostly relating to the editing and creation of new macros. (The functionality of these icons largely reproduces functionality of icons on the toolbar or options in the various menus on top.)

- **Categories Pane**, where macros may be grouped in folders by functional area, type, or other criteria. The category folders are internal to the macro file; each macro file may have its own category folders.

- **Macros Pane**: for each macro there is a listing of its nickname, activation, scope, and date of most recent modification. The macros may be sorted by any of these criteria, by left-clicking on the bar at the top of the column. Clicking on the bar a second time will reverse the sort.

Revised March 10, 2009
4.4 Understanding Individual Macro File Structure

4.4.1 Opening a Macro

To open a macro and view its structure, double-click on the macro name in the Macros Pane. For this example, please scroll down to the macro gen_UserDefinedStrings and double-click on it.

When the macro opens, the screen will display two panes: Actions, which will appear at all times, and Scripting Editor, which opens as the default macro editing environment. (There are other macro editing environments, but they will not be covered in this class.)

The Scripting Editor has five tabs:

- Script
- Properties
- Scope
- Security
- Notes

Note: Some of the menu and toolbar items in the Scripting Editor window are different from those in the Macro Explorer window.
4.4.2 Understanding the Script Tab

The macro script will open in the Script Tab which consists of two active sub-panes:

- The Command Pane displays a list of macro command groups. Each group may be expanded by clicking on the group name to show the complete set of commands associated with that group. Any known command can be accessed by typing its name into Search Box below the Command Pane.

- The Script Pane contains the body of the macro: a sequence of macro commands (in black) interspersed with remark lines (in blue). Remarks are non-functional elements that serve as guideposts mainly for macro developers and editors. Users seeking an overview of a macro’s purpose and functions should consult the Notes Tab (Section 4.4.6).

4.4.3 Understanding the Properties Tab

The Properties Tab is where the nickname, icon, activity status, and activation method may be set or changed. The right side of the Properties tab is devoted to macro Activation; the different methods will be explained in Section 6 (with detailed explanations of Hot Key, Schedule, and Shortkey).
4.4.4 Understanding the Scope Tab

A given macro may be designed to work within a variety of programs or be limited to a single application or even a single window. The Scope Tab offers four options for defining the working environment of a macro:

- **Global**: the macro will run with any active program or window on top.
- **Global Except**: the macro will run with any program open on top except the one (or more) specified. This option can be used to prevent accidental macro activation by keyboard commands intrinsic to programs like Microsoft Word.
- **Window Specific**: the macro will run only if a certain window is running or active.
- **Program Specific**: the macro can only be run in the specified program(s), or more narrowly, when a specified program is running on top.

Macro Scope tab

This macro is limited to run with specific programs open on top. To see which programs, click on the Select button next to Program Specific.

Revised March 10, 2009
In this example, the macro is set to run only if the Aleph Cataloging, Acquisitions, or Circulation program is running on top.

The left hand side of the Select Programs window lists applications that are currently running. Clicking the “Select from a list of installed programs” button will bring up a comprehensive list of programs, but be warned: it is not user-friendly; it’s easier to open the program you need then select it from the list on the left.

Close the Select Programs dialog box either from the Cancel button or from the ‘x’ in the upper right corner of the window.
4.4.5 Understanding the Security Tab

The Security Tab enables MEXperts and administrators to set passwords, restricting access to shared macros. A password or passwords may be required to run and/or edit a macro.

![Macro Security tab](image)

Macro Security tab

Revised March 10, 2009
4.4.6 Understanding the Notes Tab

The Notes Tab provides a space where the macro creator, implementer, or user, can add notes about macro functions, programming, use, related workflow, etc.

The Notes Tab is used by the unit contributing the shared macro to provide more detailed information about it.

Note: When viewing shared macros it is important to scroll down and read through all of the notes.

Revised March 10, 2009
The HUL shared macro set typically contains the following information in the Notes Tab:

- **Macro name**: a plain English name by which the macro is known in documentation outside Macro Express. (For example, the macro nicknamed gen_UserDefinedStrings is called User-defined strings.)

- **Purpose**: explains what the macro does from the end-user’s standpoint. This text also appears on the HUL Shared Macros summary spreadsheet.

- **Starting point**: the application, window, or specific functional point from which the macro should be invoked in order to run properly. (In this case, the macro may be invoked anywhere in the Cataloging, Acquisitions, or Circulation modules.)

- **Functions**: a more detailed description of the macro’s functioning.

- **Additional information**: details about implementing, invoking, or running a macro. This section may specify other macros (subroutine macros) or helper files required for proper functioning of the macro.

- **Development history**: specifies the application (and the version) for which the macro was written, identifies the person who wrote the macro, and states the date when the macro was last revised.
4.4.7 Closing the Macro and Returning to Macro Explorer

To return to the list of macros, left-click on the Macro Explorer icon in the left pane or select Return to Macro Explorer from the File menu. If you have modified the macro, you will be prompted to save your changes.

![Save Changes dialog box](Image)

(If you did not intend to make any changes, click No.) If you did not make any changes, you will be brought immediately to the Macro Explorer. To save a macro without closing it, click on the Save icon in the toolbar, select File > Save from the menu bar, or type Ctrl-S.

Close the gen_UserDefinedStrings macro and return to the Macro Explorer.

Exercise 2: Reviewing Scope and Notes Tabs

From Macro Explorer, select the acq_inv_VerifyOpenOrCreateInvoice_v2.0 macro. Determine macro’s scope – i.e., if it can run in all programs and windows or only in specific programs or windows (and if so, which) and whether this macro requires subroutine macros or helper files.

5. Importing Macros

Once you have determined which macro(s) you want to implement from the shared macro set, you will need to import them to your working file.

**Best practice:** Keep the downloaded macro file as a master file, and create a new file where you will import desired macros and where you can make local edits.

5.1 Creating a Local Macro File

If you do not already have a local macro file, you will need to create one at this time. To create a macro file, launch Macro Express, and open the Macro Explorer window. From the File Menu, select New Macro File. You will be prompted to select the location and name for the new file.
Name the macro file, and set a location on your hard disk. This will be a working file. The file that you provide to other staff members in your unit will be a copy of this file.

For this exercise, name the file “Test.”

**Best practice recommendations:** It is best to choose a local hard disk location for any macro file that will be used for creating or editing macros. Working on locally-stored macros is faster and you will avoid potential problems from editing a file that others may be using at the same time. Once you are done with your edits, you can copy the file from your local drive to a shared network location if that is your choice. It is best not to edit a macro file while it is simultaneously in use by others. As with any important files, you should keep backup copies of your macro files. We will discuss file location choices in more detail later during this class.
The resulting new “empty” macro file will actually have a few macros in it – these are Macro Express system macros and cannot be deleted.

5.2 Importing Macros from Shared Macro File to Local Macro File

1. Create a Category folder to which you can import selected macros.

   **Best practice:** Create at least one category where you will store your own macros. That way, the category named *Unassigned* will be reserved for system macros only.

   a. From the Category menu select Add Category.
   b. Type in the name of the Category (for the purposes of this class, call it “Macro Class”) and press Enter.
   c. Make sure the new Category is highlighted before beginning import steps.
Create new category
2. To import macros to the newly created Category in your target file, select: File Menu > Import > Import Macros

![Import options on File menu](image)

3. This will bring up an Import Macros window. Click on the Open File button.

![Import Macros window](image)

Revised March 10, 2009
4. This will display a list of macro files in the active folder. You may need to navigate to another folder in order to display the macro file that you want. For this exercise, open the HUL_Shared_Macros.mex file by left-clicking on the file name and then on the Open button.

![Image of Import Macro File dialog box]

Window to select macro file to open
5. A list of macros available for import from the source macro file will be displayed.

![Beginning list of macros available for importing](image1)

If the list is long, you may need to scroll down to see the macros that you want.

![The same list of macros after scrolling down](image2)

Revised March 10, 2009
6. Select the macro or macros you wish to import. Some options:

- Left-clicking on a macro name will select that macro from the list.
- If you wish to select several but not all macros, you can left-click while holding the Ctrl key down to select/deselect individual macros or with the Shift key down to select adjacent sequences of macros in the list.
- You can select all macros by clicking on the Select All button.

For the purposes of this exercise, select the following macros:

- `cat_auth_create_NameAuthority_v0.6`
- `cat_auth_create_NameTitleAuthority_v0.6`
- `cat_auth_create_SeriesAuthorityRecord_v0.6`
- `cat_auth_create_sub_CreateAuthority_v0.6`
- `cat_auth_edit_SeriesAuthority_CSS`
- `cat_auth_edit_sub_ReadMARCforHfield`

leaving out the Widener-specific macros:

- `cat_auth_create_SeriesAuthorityRecord_WID_v0.6`
- `cat_auth_edit_SeriesAuthority_WID`

The following graphic demonstrates which macros should be highlighted for import.

![Multiple macros selected for importing](Multiple_macros_selected_for_importing.png)
7. Once you have highlighted all macros that you wish to import, click the Import button. When the import is complete, you will see an “Import Complete” message.

**Note:** The more macros you are importing, the longer the process takes. It could take up to a couple of minutes if either the source file or the target file is on a network drive.

![Import Complete message]

8. Click OK, and the macros will appear in the Macro Explorer window in the currently open category folder.

**Note:** You may have to click into another category folder and back into the desired category folder in order for the macro list to display correctly.

![Imported macros]

Revised March 10, 2009
Exercise 3: Importing Macros

Create a category, name it Practice, and import the following macros from the master file HUL_Shared_Macros.mex:

- gen_UserDefinedStrings macro
- genlib_file_CopyExeOrDllToLocalDrive_vA
- genlib_SetupMacroExHCLTSSubfolder_vA

Once the macros have been imported and assigned to a category if desired, a macro activation method should be selected for each macro that will be run. This will be covered in Section 6.

No activation will be assigned to subroutine macros that are invoked only by another macro.

**Note:** When you import macros from the HUL Shared Macro file, they will not have activation keys assigned to them. However, if you import macros from a file that already has activations assigned (for example, when moving macros from one working file to another), you might inadvertently try to import a macro that has the same activation (e.g., hot key Win-Z) as a macro in your destination file. Under these circumstances, you will receive an error message:

![Duplicate Warning dialog box](image)

If you answer yes, and the macro is imported without activation, you will be able to set a new activation for the macro.

**Note:** Macro Express will allow you to import macros with identical nicknames to the same file. If you are importing new versions to replace macros in your local file, it is best to delete the old versions from the file before importing new ones.

When you delete macros from a Macro Express file, they are sent to the Macro Recycle Bin, which is part of the macro file. If you delete a macro accidentally, you can retrieve it from there: click on the Recycle Bin icon, select the macro or macros you want to keep and choose Restore Macro(s). Macros in the Recycle Bin are still part of the macro file and contribute to its size, so it is recommended that you empty the bin periodically in order to keep your file from growing indefinitely. Once you delete a macro from the Recycle Bin, it is permanently removed and cannot be recovered.

Exercise 4: Deleting and restoring from Macro Recycle Bin

Put the macro genlib_file_CopyExeOrDllToLocalDrive_vA into the Macro Recycle Bin, and restore it.

Revised March 10, 2009
6. Macro Activation Methods

The Properties Tab allows you to set the activation method for the macro and change the nickname, the activity status, and the macro icon (the icon is often used in conjunction with setting the macro activation method). The choice of an activation method is a local decision and may depend on a personal preference or the most efficient method for your local computing environment. It is also possible to have more than one type of activation assigned to any macro. Please note that if more than one activation method is assigned to a particular macro, the program will prompt you to select only one of them to display for that macro in the Macro Explorer.

6.1 Hot Key Activation

A hot key is a key or combination of keys that, when pressed together, will play back a macro. Macro Express allows hot keys consisting of a single key up to four keys pressed simultaneously.

To create a macro hot key:

1. In Macro Explorer select the macro you want to activate – in this case choose cat_auth_create_NameAuthority_v0.6 – and open it in the Scripting Editor.

2. Navigate to the Properties tab.

3. Select the hot key activation method by checking off the Hot Key checkbox, which will enable the Set HotKey button. Click on the Set HotKey button, which will open the Set HotKey Activation dialog box.

Revised March 10, 2009
4. Designate the hot key. There are two ways to do this:
   a. Key the desired keystroke combination into the Set HotKey Activation box and click OK.

   OR

   b. Scroll through the hot key list, select the desired hot key combination, and click OK. When selecting a hot key, confirm that the hot key you are selecting is not already in use by the native application your macro is working with or by the Windows system.

   **Macro Tip:** The hot keys listed in red are those that might interfere with reserved Windows hot keys. “Global Key Not In Use” refers to hot keys assigned to macros with Global Scope (macros that can run in any program). The Set HotKey Activation window does not indicate if hot keys are in use for macros with limited scope (for example, macros that can only run in the Aleph Cataloging module).

   Please consult Choosing Hot Keys documentation on the HUL Macro Express Web site, which includes lists of Aleph Keyboard Equivalents and Keyboard Shortcuts for Windows. (For example, one should not assign a hot key to Ctrl-2 for an Aleph macro, because Aleph uses that key combination for navigating to the upper right pane.)

5. Save changes to the macro.

6. Return to the Macro Explorer.

   The new hot key will appear in the Activation column of Macro Explorer and is now ready for use in the program(s) specified in the Scope column.
Exercise 5: Assigning a hot key

Assign the hot key Win-A to the macro gen__UserDefinedStrings.

6.2 Schedule

Macros can be scheduled to play back at a specific time. Just set the schedule for the macro and it will play back unattended at the designated time – no need to press a hot key or click on an icon. In order for your macro to execute, Macro Express must be running in the background and the right macro file must be open.

To schedule a macro, go to the macro’s Properties tab, check off the Schedule activation method, and click on the Set Schedule button. That will bring up the Set Schedule dialog box, which gives you the following scheduling options:

Scheduled macros are not commonly used in HUL. An example of one in use is a macro that opens Aleph at machine startup and pauses at the login screen; another is one that empties the Windows Recycle Bin periodically.

6.3 ShortKey Activation

A ShortKey is a sequence of characters such as abc, myname, etc. When the characters are typed into an application, the macro runs. A ShortKey can be from two to ten characters in length. They are preceded by a prefix which is a signal to Macro Express that the macro should be triggered.

Macro tip: ShortKeys are good for macros that input commonly-used text strings and search commands. They allow users to invoke macros without breaking their keying.

To create a macro ShortKey:

1. Select the macro for which you would like to assign a ShortKey – in this case cat_auth_create_NameTitleAuthority_v0.6 – and open it in the Scripting Editor. Select the Properties tab.

Revised March 10, 2009
2. Fill in the ShortKey checkbox and type the desired ShortKeys in the box below it. (In this example, “nt” has been entered as the ShortKey for the macro in question.)

3. The Settings button allows you to designate ShortKey settings that apply only to the current macro. In most cases, this step will not be necessary. (For an explanation of these options, see the Macro Express Help documentation.)

4. Save changes to the macro. The new ShortKey will appear in the Activation column of Macro Explorer and is now ready for use in the program(s) specified in the Scope column.

Revised March 10, 2009
**Note:** The ShortKey characters will disappear when the macro is fully invoked – if you use this in Aleph and type ',,' the commas will appear on the record, but as soon as you complete the ShortKey and type 'nt' the commas and the ShortKey combination 'nt' will disappear.

5. The Macro Express Version 3.7d installation at Harvard comes preset with ‘,,’ (comma-comma) as the Global ShortKey invocation prefix, but if a different prefix is desired, it can be changed in the Global ShortKeys settings: open the Options file menu, select Preferences, and click on ShortKeys. The Preferences window includes global options for invoking a macro with a ShortKey.

A ShortKey can be invoked using either a prefix or a suffix. Note, however, that the two activation types cannot be used concurrently on a given computer: as with all global options the ShortKey setting will apply to all macros run on the local computer. It is a good idea to choose a ShortKey prefix that is easy to key and constitutes a sequence of characters that ordinarily wouldn’t appear independently when typing. (The Suffix option will not be covered in this class.)

![ShortKeys Global/Default Preferences window](image)

This page provides other settings options in conjunction with ShortKeys; for an explanation of these options, see the Macro Express Help documentation.

### 6.4 Popup Menu

A popup menu is a menu of macros you have previously created. Once created, it appears as a macro in the list of macros in the Macro Explorer. The menu window can be opened by pressing a hot key of your choice or clicking on a system tray icon. To create a macro popup menu:

1. When in Macro Explorer, select the Category Macro Class.
2. Go to the Macros menu and select Add Macro.
3. Select the Popup Menu radio button activation option, and set the hot key for the popup menu in the Select a Hot Key box. For this exercise use the hot key: Win-Z.

Creating a Popup Menu

**Macro tip:** Two macros can share a single hot key activation (for example, you can set Win-Z as the hot key activation for two different macros) but unless the scope of each macro is set to run in different program environments, double-assignment will cause interference, and only one of the two macros will run. In this case, you will need to change the key assignment for either the individual or popup menu macro.
4. Click on the Build Menu button, which will launch the Popup Menu Builder, including an Available Macros list from which you can populate the popup menu.

5. Highlight multiple macros on the Available Macros list to be added to the popup menu and use the Add arrow to add them to the popup menu list of macros. Alternatively, you may wish to highlight a single macro, and use the Insert or Add arrow; Insert will add the macro to the highlighted point on the popup menu macro list. Add will place the selected macro at the end of the Macros in Menu list.

For this exercise, add the following macros to the menu:

- `cat_auth_create_NameAuthority_v0.6`
- `cat_auth_create_NameTitleAuthority_v0.6`
- `cat_auth_create_SeriesAuthorityRecord_v0.6`
6. The order of macros in the menu can be adjusted afterwards by selecting a macro and using the Up and Down arrows on the right hand portion of the window.

Note: Macros may be added, inserted, and removed from the existing menu list at any time after the menu is created.
7. In the Properties tab enter a descriptive Nickname for the popup menu – in this case, Authority Record Creation:

8. If desired, a persistent icon for the popup menu can be placed in the system tray by filling the checkbox for Place menu in System Tray in the Properties tab. If this option is used, a hot key is not necessary to open the popup menu.
9. Also, in the Properties tab select display options for the popup menu (Icons Only, First Part of Macro, or Nickname and Scope). For this exercise, we will choose First Part of Macro as the display option.

Each option has its pros and cons:

a. The Icons Only option provides the most compact and visually appealing menu layout but is not very descriptive. It may not be easy to find an icon that will be easily identifiable with a macro. And for long menus (they can have up to 36 entries), the Icons Only option may be confusing. The Icons Only option will be demonstrated when setting up a floating menu.

b. Nickname and Scope – With this display option, the popup menu appears in list form, displaying the contents of the Nickname field. This is a good choice for your popup menu label when the Nickname is something clearly and easily understandable.

When you are writing your own macros for in-house use, this option is fully available to you. When using macros taken from the HUL Shared Macro Set, you may find that the names that come with the macros, based on the HUL naming conventions, are not optimal for popup or floating menus (cat_auth_create_NameTitleAuthority_v0.6) even if they are understandable. It is possible to change the Nickname of a macro, by going to the Nickname field on the Properties tab and changing the nickname that appears there. A potential drawback to this approach is that some macros have embedded in their script the names of other macros that they call upon; if the name of a “called” macro has changed, it will no longer be recognizable to other macros. You may decide to make a copy of the macro in question, and change the nickname of the copy while keeping the original in the macro file. However, if the macro is updated in the HUL Shared Macro Set, you will need to remember to replace both copies. Another option would be to create a new macro that runs the macro.

c. First Part of Macro – This is an alternate method, which is less visually pleasing, but does not have the same renaming risks as Nickname and Scope. The First Part of Macro option allows you to choose a display name for your macro, without changing the official nickname. The display name is added as a Remark line at the very beginning of the macro with a concise, straightforward name that does not need to follow the macro nicknaming convention. The disadvantage to this approach is that the Macro Express coding for remark lines will appear at the beginning of each line in the popup menu (see the figure below). Also, if you change anything in a shared macro, you must make the same changes again when you import a revised version of the macro from the shared file.

10. Save changes to the popup menu macro, and then return to the Macro Explorer window by clicking on the Macro Explorer icon in the Actions pane at left.

The new popup menu macro will be visible in Macro Explorer and is now ready for use.

11. To activate the popup menu use the assigned hot key, or if an icon was placed in the system tray click on the popup menu icon.

12. To activate a macro from the popup menu, you may double-click on it, single-click on the macro name and press Enter, or key the number/letter to the left side of the macro.
Macro tip: Macros that assume that a particular window (such as Aleph Cataloging) is open on screen will not work if you run them from a popup or floating menu. If you wish to run such a macro from a popup or floating menu, you will need to add an Activate Window command to the beginning of the macro to activate the desired window (such as Aleph Cataloging).

Exercise 6: Creating a Popup Menu

- Import the following macros from HUL_Shared_Macros.mex to test.mex:
  - cat_auth_create_NameAuthority_v0.6
  - cat_auth_create_NameTitleAuthority_v0.6
  - cat_auth_create_SeriesAuthorityRecord_v0.6
  - plus any subroutine macros required by these macros.
- Create a popup menu called “Authority records” with a hot key activation (assign it to Ctrl-Z)
- Under Properties Tab, select “Nickname and Scope” under How to Display and change hot key setting to Win-Q
- Add these three macros to the popup menu.
- View the popup menu by pressing Win-Q.

6.5 Floating Menu

6.5.1 Creating a Floating Menu

A floating menu has the same form and options as a popup menu except that it always remains on top and is always accessible as long as the Macro Player is running. A floating menu can be minimized to just the caption bar to free up space if needed. Multiple floating menus can be created, but only one can be run at a time. The last screen location of the floating menu is retained and used with subsequent runs of Macro Express.

A floating menu can be created in much the same way as a popup menu:

1. When in Macro Explorer, go to the Macros menu select Add Macro.
2. Select the floating menu radio button and click on the Build Menu button. *(Note: HotKey activation is not an option like it is for the popup menu, as a floating menu is always onscreen, unless it has manually been deactivated.)*

The Floating Menu Builder works in the same way as the Popup Menu Builder shown in the previous section. In the Editor tab select macros to add to the menu and in the Properties tab create a Nickname and choose menu display options.

We will not create a new floating menu here. Instead, we will transform an existing popup menu into a floating menu. Therefore, close the menu builder by clicking on the Macro Explorer icon in the Actions column, and respond No to the “Do you want to save changes?” dialog.
6.5.2 Changing a Popup Menu to a Floating Menu

1. In the Macro Explorer, right-click on the popup menu that you wish to transform to a floating menu and select Properties from the right-click menu.
3. This brings up the menu’s Properties window. Note that it is NOT the same as the Macro Properties tab. (This window can also be opened from the menu bar: Macros > Properties.)

Menu Properties window

4. Change the radio button from Popup Menu to Floating Menu and click on OK. The floating menu will now be visible at the upper left of your screen and will appear in the Macro Explorer list with the term “floating menu” in the Activation column.

**Macro tip:** You can move a floating menu to a different location onscreen by dragging it by its title bar. The new location will be “remembered” next time Macro Express is launched.
5. You can now go into the Properties tab of the Authority Record Creation floating menu and change the display option if you wish. In this case we will change it to Icons Only.

Floating Menu before changing display option to Icons Only

6. Save changes to the floating menu macro.
6. A macro is activated by clicking on its icon on the floating menu. Note that if all the macros on the floating menu have the same icon assigned to them, the resulting floating menu will be a series of identical icons, which is clearly not useful.

Examples of Floating Menus with identical icons

Section 6.6 below offers detailed instructions on changing macro icons. You may wish to change a macro’s icon before adding it to a floating menu.

**Macro tip:** Macros that assume that a particular window (such as Aleph Cataloging) is open on screen will not work if you run them from a popup or floating menu. If you wish to run such a macro from a popup or floating menu, you will need to add an Activate Window command to the beginning of the macro to activate the desired window (such as Aleph Cataloging).

7. Floating menu can be closed (or deactivated) and opened (or re-activated):

a. Closing the floating menu from the ‘x’ in the upper right corner will deactivate the menu. The deactivated menu macro will appear in Macro Explorer in italics.
b. To reactivate the floating menu, right click on the floating menu name and select Run Macro Now or check off Active on the macro’s Properties tab.

Exercise 7: Creating a Floating Menu

- Modify your popup menu “Authority Records Creation” to make it a floating menu.
- Change the How to Display option on Properties tab to Icons Only.
6.6 Assigning an Icon to a Macro

You may minimize the amount of space a floating menu takes up on-screen by having an icon, instead of a macro name, display in the menu. However, all macros have the same default icon: and in the HUL shared macro set, identical icons have been assigned to a group of macros to sort them, as we saw above with all A’s for the three authority record creation macros.

1. To differentiate macros on a floating menu with individual icons, open the macro in question in the Macro Explorer and go to the Properties Tab.

   ![Change Icon button in the Macro Properties tab]

2. Click on the Change button in the Icon section to bring up the Select an Icon dialog box.

   ![Select an Icon dialog box]
3. When the Select an Icon box appears, click on the Browse button. This will default to the Macro Express Icons Library folder.

(Note: If a different folder opens, navigate to the Macro Express icons folder at C:\Program Files\Macro Express3\Icons\ -- you may also use icons that you acquire from other sources or create yourself using a graphics program.

4. Review the icons and choose the one you want. In this case, click on the green “N” then click the Open button.)
5. When the Select an Icon window re-appears, click on OK.

![Select an Icon dialog box]

6. Now the new macro icon will appear on the Properties Tab.

![Macro Properties tab with changed icon]

The icon on the floating menu on screen will change accordingly.
7. In the same manner, you can replace the icon for `cat_auth_create_NameTitleAuthority_v0.6` with a red “T”: and for `cat_auth_create_SeriesAuthorityRecord_v0.6` with a blue “S”. The resulting floating menu will look like this:

![Floating menu with differentiated icons](image)

Exercise 8: Assigning icons to a macro

Assign the following icons to these macros:

<table>
<thead>
<tr>
<th>Macro</th>
<th>Icon</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>cat_auth_create_NameAuthority_v0.6</code></td>
<td>green “N”</td>
</tr>
<tr>
<td><code>cat_auth_create_NameTitleAuthority_v0.6</code></td>
<td>red “T”</td>
</tr>
<tr>
<td><code>cat_auth_create_SeriesAuthorityRecord_v0.6</code></td>
<td>blue “S”</td>
</tr>
</tbody>
</table>

(Note the change in the appearance of the floating menu.)

6.7 Other Activation Methods

Three additional macro activation options will not be covered in this class:

- **Window Title:** on the appearance of a window with a particular name
- **Mouse Click:** upon the click of a mouse at a specific screen or active window coordinates
- **Control:** upon the activation of a particular Windows control (button, box, etc.)

None of these activation methods are currently in wide use within HUL to the knowledge of the Aleph Macro Working Group. If you would like to pursue using one of these methods and need help, please consult with the Aleph Macro Working Group.

**Note:** In addition to the assigned activation methods, there are two other methods of running a macro. The following two commands can be used even before assigning an activation method to a macro and are primarily used when writing and testing a macro with the Macro Express program active and open.

- **Run Macro Now:** From Macro Explorer, either right-click on the macro you want to run and select Run Macro Now or, with the macro highlighted in Macro Explorer, select Run Macro Now from the Macros file menu.

- **Test Run Macro:** With the desired macro open in the Scripting Editor, either select Test Run Macro from the Debug menu, or press the F9 function key, or click on the green triangle icon on the toolbar. This command can be used even before saving the macro.

Revised March 10, 2009
7. File Management

Like any files, macro files and related helper files (which are explained below, in 7.2) can be stored on a local computer or on a server; files on a server can be stored on a personal network drive, or on a shared network drive. These different locations have different implications for the management of macros and macro files, and each choice has advantages and disadvantages for the MEXpert.

7.1 Macro File Location

MEXperts need to decide on the best location for macro files that will be used by staff in their units. Under some circumstances a macro file will be used and maintained by only one staff member, in which case placing the file on the user’s C: drive may be the best option. When several staff members share a macro file, however, it can be impractical to store the file on each individual’s C: drive. The file must be copied to each computer, and any changes to the file (additions or revisions) will need to be made on every machine—a time-consuming and often unnecessary multiplication of labor for the unit’s MEXpert (unless you have an automated way of writing to the individual C: drives). In the case of multiple users, therefore, it may be preferable to store the macro file on a shared drive from which it can be accessed simultaneously by several people.

If a unit decides to run macros from a shared network drive, it is recommended that any macro revisions or file maintenance be done on a local drive – in other words, the MEXpert should keep a copy of the macro file on the C: drive, edit the local file, and then copy the revised file to the shared drive or import the individual revised macros into the macro file on the shared drive. It is good practice to inform users whenever macros or files are being replaced.

For added security – primarily to prevent accidental alteration or deletion of shared macros – limited access privileges (security authorizations) may be considered for shared macro files. For example, Read/Write access could be given to only the MEXpert and perhaps one or two backup persons, and most staff could be granted Read access only.

Regardless of where it is located, any macro file should be backed up to a shared drive or other location in case of machine failure or file corruption.

7.2 Location of Helper Files

Certain macros in the HUL shared macro set use text files and other helper files for various purposes. Some of these files are temporary: the macro creates, deletes, and overwrites them as needed. Others are more permanent in nature: they must already exist at the expected location when the macro is run. The documentation in the Notes tab of each macro includes information on which helper files, if any, the macro uses and where it looks for them. The path and file name of associated helper files are specified in the macro script, which must be edited if the location or name of the file changes. Appendix D explains different types of files currently in use.

7.2.1 Temporary helper files

The temporary files that a macro creates, deletes, and overwrites are typically saved in the Macro Express program directory on the C: drive. It is assumed that throughout the various libraries and units, the program is installed at C:\Program Files\Macro Express3\.

Some macros in the HUL shared set use helper files to keep track of saved settings or preferences for those macros, such as staff initials used by some units in acquisitions notes. These files are created and edited by the macros that use them, and typically saved on the I: drive or (if there is no I: drive) in the Macro Express program directory on the C: drive.

Revised March 10, 2009
The location of these files is usually not an issue, but if error messages come up, you should verify that the file paths that the macro uses are valid in your environment and that the staff have Read/Write access to those locations.

### 7.2.2 Preloaded helper files

The helper files that require the most attention in this discussion are those that must be preloaded in a specific folder before running the macro. These are look-up files that store information that macros use from time to time, such as a list of MARC language or country codes, or small external program scripts that some macros run. Before running a macro that uses preloaded helper files, you must choose where to store these files. Then you must download and copy the files to that location and make sure that the macro knows to look for the files in that location.

Just like macro files, helper files can be stored on a local computer or on a network drive. If you choose a folder on a shared network drive, you only have to copy the files to that one location. But if you choose a folder on a local computer or on a personal network drive, you must copy the helper files to each user’s computer or personal drive.

The information in these files is common to all users, so it makes sense to store them in one location accessible to all staff in the unit. The default path for these files, as specified in the macro script, is either the Y: drive or a subfolder on it. (The Y: drive is a mapped network drive, which is usually set up once for each user.) If you choose a different location for these helper files, other than the default Y: drive path, you must change the macro script to refer to the new location. (If you do this, it is a good idea to add a remark line in the script documenting the change, so it will be easier for you to go back to it later if needed.) Whatever your decision, the macro script must match the actual location of the files.

If you choose to keep the default Y: drive path because you would rather not edit the macro script, you have to choose a folder on a network drive and map that folder as the Y: drive. Drive mapping is a one-time step on each user’s network login on each computer that the person uses. Appendix C contains instructions on how to map a network drive.

When using a folder on a shared network drive, you should make sure that everyone in your unit has the appropriate file permissions for the selected network location. You may consider setting up read-only permissions for most staff to reduce the chance of accidental deletions or changes.

Please note that the choice of the helper file location is independent of the macro file location. You can have your macro file on the C: drive but let the macros refer to helper files on the Y: drive, for example. Or, you can place your unit’s shared macro file and all associated helper files on the same network drive.

---

Macro files in the Macro Express program folder on the C: drive

Revised March 10, 2009
Helper files and macro files in a network drive folder mapped as Y: drive

Example of references to helper files in a macro script (1)

Revised March 10, 2009
Example of references to helper files in a macro script (2)

If you need help setting up the access permissions to shared network folders or with Y: drive mapping, contact your IT department. If you would like guidance in selecting the location that best suits your local environment, feel free to contact the Aleph Macro Working Group.

Exercise 9: Change path for helper file.

You have decided to implement the Reference Lookup macro that allows you to look up MARC language or country codes and to select a code to type at cursor position. You have also decided to store the helper files this macro needs, not on the Y: drive, but in the Temp directory on the C: drive.

Import the macro gen_lookup_ReferenceLookup macro into your local file and open it. Next, find the first “Variable Set String” command and change the path from Y: to C:\Temp (be sure not to edit the name of the file, only the path). Find the other occurrences of the Y:\ path and change them as well. Save the macro.

(If you actually implement this macro in your own unit, make sure to read the full documentation in the Notes tab. There you will learn, among other things, that you will also need the macro gen_lookup_sub_ReferenceLookupTable.)
8. Getting help

8.1 Aleph Support Center and Documentation Website

A copy of this document can be found on the OIS Aleph Documentation Website in the Macros Section at http://hul.harvard.edu/ois/systems/aleph/docs/train_mexa.pdf

You can also contact the Aleph Support Center for:

- Questions about the HUL Shared Macro Set
- Suggestions for additional macro training or documentation.
- Submission of macros that you have created for inclusion in the HUL Shared Macro Set

The Aleph Support Center is located at: http://hul.harvard.edu/ois/systems/aleph/support.html

Select the Macro Express category to send questions, feedback and other requests to the Aleph Macro Working Group. Please note, however, that support for macro programming problems is not available from the Aleph Support Center.

8.2 HUL Macro Express iSite resources page

The HUL Macro Express iSite resources page has a variety of documentation to assist in macro use and development. See also the Tips section on the documentation page for information on best practices and specific problems. You are encouraged to submit tips of your own. The HUL Macro Express iSite is located at http://isites.harvard.edu/icb/icb.do?keyword=k1846

8.3 HUL Macro Express Users’ Group (MUG)

The HUL Macro Express Users’ Group list provides an online peer community where you can ask questions and solicit macro programming tips (mug@hulmail.harvard.edu). This group also meets every two to three months, providing regular opportunities for discussion. You can subscribe to the MUG discussion list by going to: http://hul.harvard.edu/resources/hul_lists.html.

8.4 Macro Express Software Built-In Help and Tutorial

The Help file in Macro Express software is extensive, and topics can be searched by keyword or browsed through an alphabetical list. The Help menu also gives access to a built-in or online tutorial and a keyword-searchable knowledgebase on the company’s web site. Last but not least, every command dialog box also contains a context-sensitive Help button that opens the relevant page in the Help file.

8.5 Macro Express Website

The website of the makers of Macro Express, Insight Solutions, has a number of useful resources:

- Homepage: http://www.macros.com/
- Support: http://www.macros.com/support.htm
- Tutorial: http://www.macros.com/tutorial/ME3Tutorial.htm

8.6 Macro Express Book

Those seeking to do significant macro programming to support the workflow needs of their unit may find it worthwhile to obtain the book Macro Express Explained, by Joseph Weinpert (Victoria, Canada: Trafford, 2004), which provides detailed and clear explanations of all of the Macro Express commands.

Revised March 10, 2009
8.7 Macro Express Online Communities

You may also wish to consider joining one of the Macro Express related online communities, such as the Macro Express Forums hosted by PGM or the Aleph Macro Express discussion list.
Appendix A: System Requirements for Macro Express Use

To access and use the HUL Macro Express iSite, you will need the following installed on your PC workstation:

- Web browser (Firefox, Internet Explorer), etc.
- Microsoft Excel (or another program that can read MS Excel spreadsheets)
- Adobe Acrobat Reader

To access and use shared macros or to create your own, you will need the following installed on your PC workstation:

- Macro Express (the version currently distributed by OIS)
- ALEPH or whatever application(s) with which you want to use macros.

Appendix B: Macro Naming Conventions

Note: The current HUL shared macro set follows a set of naming conventions for macro “nicknames”, which will be explained below. For reasons of historical development of the macros, the names of macros in the current set are not always 100% compliant with these naming conventions. During an upcoming upgrade of these macros to the next version of Macro Express, all macro names will be brought into compliance.

Unicode character macros (see end of macro list) can be used for typing pre-composed characters (letter-diacritic combinations), special characters, and diacritical marks, and have names based on the Unicode characters they type.

Other macros in the shared set will in most cases have names based on their functional area. The naming convention used for shared macros is as follows:

a) Aleph names start with functional area prefix – non-Aleph applications start with an application prefix.

The first segment of an Aleph-related macro name will be a brief code designating the functional area for which the macro was developed. Prefixes currently in use are “acq” for acquisitions, “cat” for cataloging, and “gen” for “general ALEPH functionality.” For macros designed for other applications, such as OCLC Connexion and Olivia, this segment may consist of an abbreviated form of the application name or be omitted altogether.

Prefixes currently in use are:

- acq_ Acquisitions
- cat_ Cataloging
- gen_ General ALEPH functionality
- Cx_ OCLC Connexion
- olivia_OLIVIA

Prefixes will be established for other functional areas (in Aleph), as relevant macros are added to the shared macro set.

General subroutine macros run by multiple macros are named starting with the prefix “genlib”.

Revised March 10, 2009
Subroutine macros designed to be invoked by a specific group of macros may be named so that
the names alphabetize with the group, adding the segment “sub” after the other prefixes.

**Note:** Some of the older shared macros do not follow the above conventions consistently. These
naming inconsistencies will be cleared up in the course of the migration to the next Macro
Express version.

b) In many cases there will be a second segment of the name that is used to designate a code for
the module, sub-module area, or general area of functionality where the macro is commonly
used, e.g. cat (Cataloging), acq (Acquisitions), cir (Circulation), inv (Invoice), fund (Fund).
Macros written for other applications may start with this segment.

c) The third component of the macro name describes the macro functionality in a concise and
descriptive manner.

d) For macros that can be localized for a particular library, there will be a segment that may be
used to designate the library code of the local library once the macro has been modified for
local use. When generally distributed, this segment will appear as “xxx”. This indicates that
the macro must be localized and the “xxx” should be replaced with a local library code. (We
recommend that libraries use their Aleph sublibrary code but local coding conventions are
acceptable and may be used to distinguish between departments or subunits.)

e) An underscore will be used between the module name, descriptive name, library designation,
and version number.

For example, the macro acq_inv_VerifyOpenOrCreateInvoice_v2.0 is Version 2.0 of an
acquisitions macro, used in the invoice tab of the ALEPH acquisitions module, which opens a
specified invoice, or creates it if it does not already exist.
Appendix C: Mapping Y Drive

Drive mapping is a one-time step on each user's network login. Because it is profile-based and not machine-based, staff must be logged on when making any drive mapping changes.

The actual steps involved in drive mapping vary slightly depending on the network operating system (NOS) that your library uses on their shared server and on the disk operating system that you run on staff computers. Below are detailed instructions on drive mapping on two commonly used NOS’s. Please contact your IT department to find out what NOS your library uses.

You will also need to know the name of the network share on the server where your selected shared folder is. Your IT department will be able to help you with that as well.

---

**Server: Windows-based network operating systems**

**Local computer: Windows 2000 or XP**

1. Open the Windows Explorer (shortcut <Win>E from anywhere).
2. Open the Tools menu.
3. Select Map Network Drive...
4. In the first pull-down menu Drive, type or select Y for the drive.
5. In the second pull-down menu Folder, type in the path for the network folder you selected for the helper files.
   
   Example: `\Hcl-wid1\Groups$\...`

   where `Hcl-wid1` is the server and `Groups$` is the share.

   **Tip:** You can copy this path from the file manager or from the properties of your selected network folder.
6. Make sure that Reconnect at logon is checked.
7. Then click on Finish.
8. The system will then connect the local computer to the network path, and a separate file manager window will open up displaying the Y: drive content.
9. If you get an error message that the computer cannot find the network path, check your spelling and/or verify that your network connection is working.
10. There is no need to log out or restart the computer.

---

**Server: Novell network operating systems**

**Local computer: Windows 2000 or XP**

1. Open the Windows Explorer (shortcut <Win>E)
2. Navigate to the network path where your selected folder is located.
3. Find your selected folder and Right-click on it.
4. Select Novell Map Network Drive from the menu.
5. In the first scroll box, choose drive letter Y:
   
   - The network path box will fill in automatically with the path to the folder.
6. Check the **top two** check-boxes only as follows:
   
   - Check to make folder appear as the top most level
   - Check to always map this drive letter when you start Windows
     
     *(Note: The third check-box should be left blank.)*
7. Press the Map button.

Revised March 10, 2009
## Appendix D: Types of text file (helper files)

<table>
<thead>
<tr>
<th>Kind of File</th>
<th>Drive</th>
<th>Purpose</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Settings Files</td>
<td>C:\ or I:\</td>
<td>• Keep track of saved settings or preferences for a macro or group of macros.</td>
<td>• If the file is missing, the macro will create it, sometimes prompting for values</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Will usually (though not always) have an .ini extension</td>
</tr>
<tr>
<td>Temporary Work Files</td>
<td>C:\</td>
<td>• Temporarily keep track of information</td>
<td>• These files are created as needed.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Help Macro Express get information out of Aleph</td>
<td>• The macro will delete any previous versions of these files before creating a new one (So, it is a good idea to empty your Recycle Bin every once in a while)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Allow Macro Express and Visual Basic applications to exchange information</td>
<td></td>
</tr>
<tr>
<td>Look-Up Files</td>
<td>Y:\</td>
<td>• Used as look-up tables to store information.</td>
<td>• On the Y:\ drive because the information is common to all users, and to make accidental deletions or changes more difficult</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Allow for simpler code</td>
<td>• File must already exist at the expected location when macro is run</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Make it easier to see what information the macro is using</td>
<td>• Will usually have a .txt or .csv extension</td>
</tr>
</tbody>
</table>

Revised March 10, 2009