AppleGlot 2.0 Quick Start Instructions

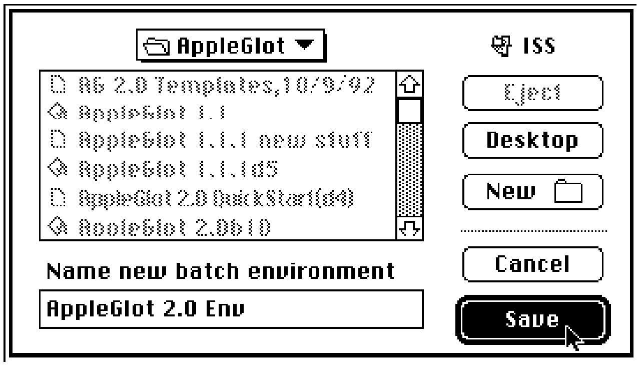
AppleGlot is an application that aids in the text translation portion of the software localization process.

These instructions are aimed at the user that is familiar with earlier versions of AppleGlot.

- Appendix A is a summary of the feature improvements from AppleGlot 1.0 to AppleGlot 2.0.
- Appendix B is a detailed description of the AppleGlot *Environment*.
- Appendix C is a detailed description of the AppleGlot Work Glossary File.
- Appendix D is a detailed description of the AppleGlot History File and Batch History File.
- Appendix E contains a list of 'TMPL' resources used for extracting strings that are built in to AppleGlot.

STEP 1: CREATE THE REQUIRED ENVIRONMENT

AppleGlot 2.0 requires a specific set of folders consisting of an *environment* folder containing a set of specifically named file folders. You may manually create this *environment* using the Finder™, however, it is much easier to create this *environment* automatically by selecting "Newenvironment..." in the "File" menu. You will be asked to identify the destination for the *batch environment* folder using a StandardFile dialog;



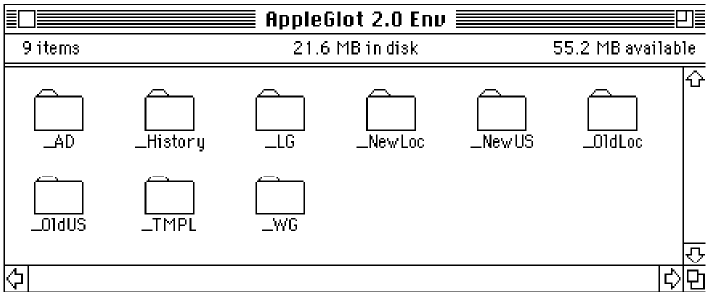
Name new batch environment

AppleGlot 2.0 Env

Generally speaking, a new *environment* should be created for each project. To avoid confusion, ***<u>don't</u>*** try to reuse old environments!

STEP 2: POPULATE THE ENVIRONMENT

The *environment* folder will contain the following folders:



Once the environment has been created, the Finder is ideally suited to populating the folders. Thus, facilities to provide this function were not written into AppleGlot. Use the Finder to move files into the proper folders. Appendix B describes the environment and the contents of each of the folders.

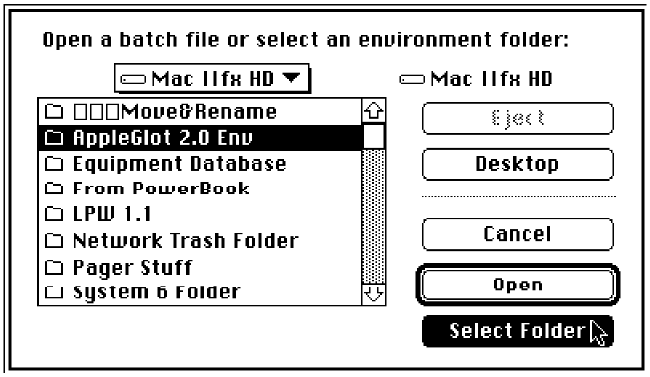
If you are localizing a new piece of software, you need only move that piece of software into the _NewUS folder within your *environment* folder. If you are localizing a new version of a previously localized piece of software, you should move the last U.S. version into the _OldUS folder and the accompanying localized version into the _OldLoc folder.

Multiple individual pieces of software may reside within a single *environment* folder. AppleGlot automatically (in almost every case) will associate the correct files using type and creator attributes.

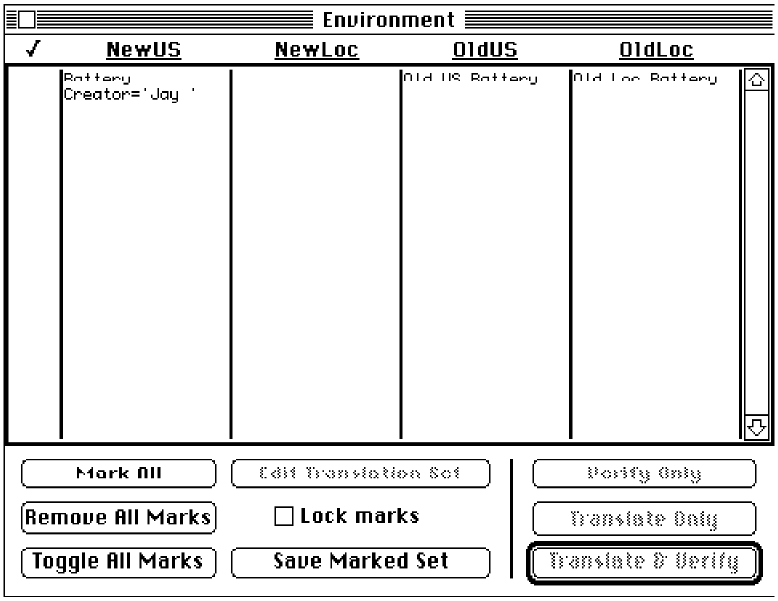
As examples for this document, two control panel files: "Labels" and "StartupDisk" have been moved into the _NewUS folder.

STEP 3: OPEN ENVIRONMENT

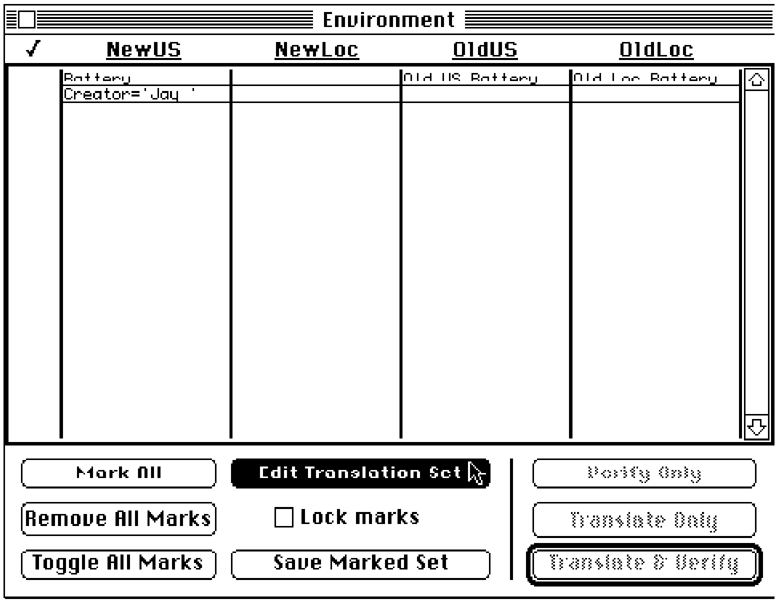
Return to AppleGlot. Select "Open..." in the "File" menu and open the environment you created and filled with files. You have no "Batch" files at this point, so the only choice is to select a folder (do not open it) and click on "Select Folder".



The AppleGlot *Environment Window* will now appear.

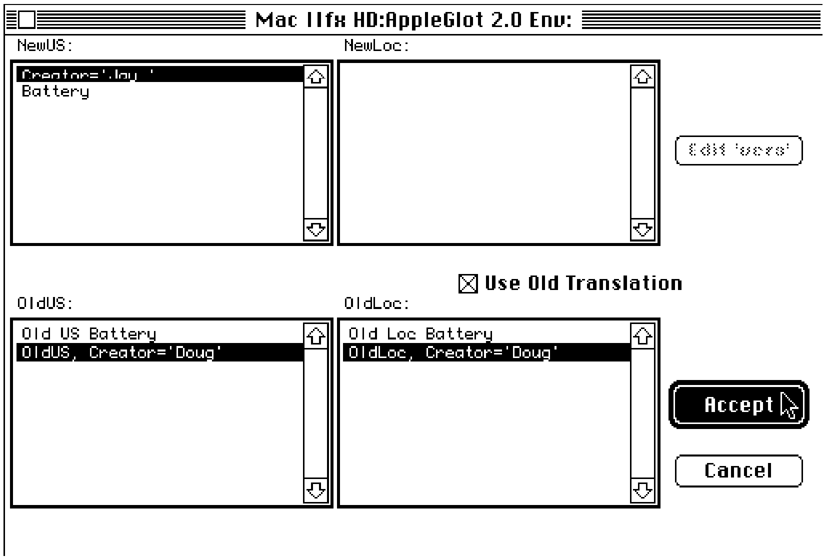


At this point, The NewUS column will show the entire contents of the _NewUS folder. AppleGlot will fill in the remaining columns through auto-association based on type and creator information and/or information contained within the batch file. Files that were not auto-associated are not yet displayed in this window. The example window above shows matching files for the "Battery" file, but not for the "Creator= 'Jay'" file due to such file creator differences. In order to include any of these files that have not been auto-associated, you must select a file in the NewUS column, and click on Edit Translation Set as seen below(first be sure that Lock marks is not checked):



✓	NewUS	NewLoc	OldUS	OldLoc
	Battery		Old US Battery	Old Loc Battery
	Creator='Jay '			

Once you have clicked on Edit Translation Set for editing the *Translation Set,* you will see the following window appear:



NewUS:
Creator='Jay '
Battery

OldUS:
Old US Battery
OldUS, Creator='Doug'

OldLoc:
Old Loc Battery
OldLoc, Creator='Doug'

Once you have highlighted the set of files that should be associated, simply click the *Accept* button. The environment window will be updated with you new associations. It is important to note that editing the translation sets will not be a common occurrence. More often than not, creator/type matches will perfectly associate the files.

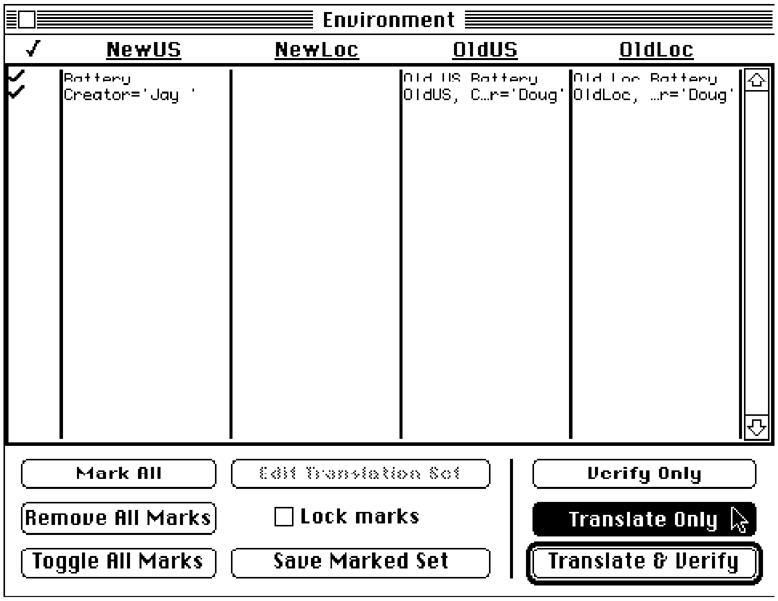
STEP 4: CREATE BATCH FILE and CLICK ON 'TRANSLATE'

AppleGlot 2.0 does all processing in *batches* even if the *batch* consists of a single file. As you click in the √ column, or *press* the buttons that deal with "Marks", check marks will appear or disappear in the left hand column of the *Environment* window. These check marks are used to designate which files from the *Environment* folders are to be grouped together into a *Batch*.

"Lock Marks" is included as a safety feature. It prevents making accidental changes to the left hand column of the *Environment Window.* You must unmark this item to be able to *mark* the files you wish to translate.

Marking files in the batch gives you the choice to translate any combination of files contained in your *Environment*.

After you have marked your files for translation, click on “Translate Only” to start the text extraction process:



✓	NewUS	NewLoc	OldUS	OldLoc
✓	Battery		Old US Battery	Old Loc Battery
✓	Creator='Jay '		OldUS, C…r='Doug'	OldLoc, …r='Doug'

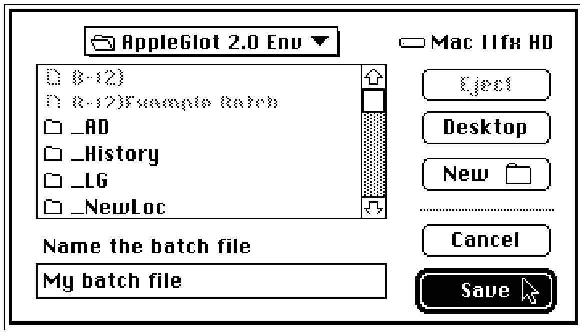
We recommend that you always create new environments for each project. So unless you have a great number of files in your project, you will probably wish to translate all the files at once. If this is the case, select “Mark All” in the above dialog. This will place check marks beside each file in the Environment Window, marking the whole set for translation. Now you are ready to click on “Translate Only” to start the text extraction process that creates all of your Work Glossaries.

You may also wish to work with your files in several smaller batches. If this is the case, simply click in the check mark column next to each file you wish to include into a your batch file. A check-mark will appear to the left of the filename signifying it marked and ready for translation.

To save time, it is recommended that you always use “Translate Only” on the first pass. “Translate & Verify” may be used on subsequent passes as the verify will provide important comparison information between the US and localized files. Also, as in the 1.x AppleGlot versions, the Verify step is required to properly set the translated file’s type and creator.

“Save Marked Set” can be used at any time to automatically save your marked files in a *Batch* file. When you click on “Translate & Verify”, “Translate Only” or “Verify Only”, you are always asked to save your *Batch* file (the current set of marked files in the Environment Window), unless you initially opened a previously existing Batch file, and have made no changes to it.

After selecting “Save Marked Set” you are automatically asked to save your *Batch* file. “Translate & Verify”, “Translate Only” or “Verify Only”, you are automatically asked to save your *Batch* file if there has been a change.

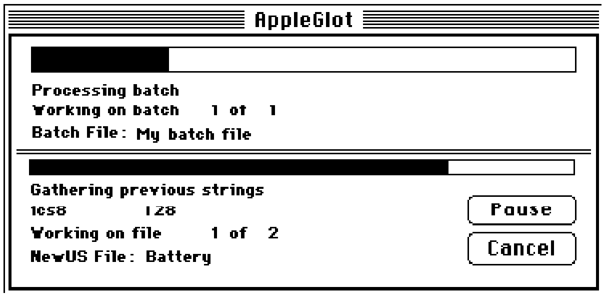


Name the batch file

My batch file

AppleGlot will first create a *Batch* file containing the paths for the set of files you selected for translation (the *marked set*). Then, using this *Batch* file, AppleGlot will perform the translations and/or verifications, as requested by the user.

AppleGlot reports its progress as the translation operations are performed:



```
Processing batch
Working on batch    1 of  1
Batch File: My batch file

Gathering previous strings
1058          128
Working on file     1 of  2
NewUS File: Battery
```

NOTES:

- **AppleGlot does <u>not</u> perform many safety checks for the user. Therefore, it is most strongly urged that you take care <u>NOT</u> to include a specific file in two different batches. It may produce unexpected results.**
- ***Batch* files contain hard-coded paths. If you, the user, rename, delete or move anything associated with the *environment*, the paths in the *Batch* file will no longer be valid! In this case, the old *Batch* file must be discarded and a new *marked set* re-saved in a new *Batch* file.**

<u>STEP 5: TRANSLATE TEXT</u>

Use your favorite text editor (such as the MPW shell) to translate the text in the work glossary files (found in the _WG folder).

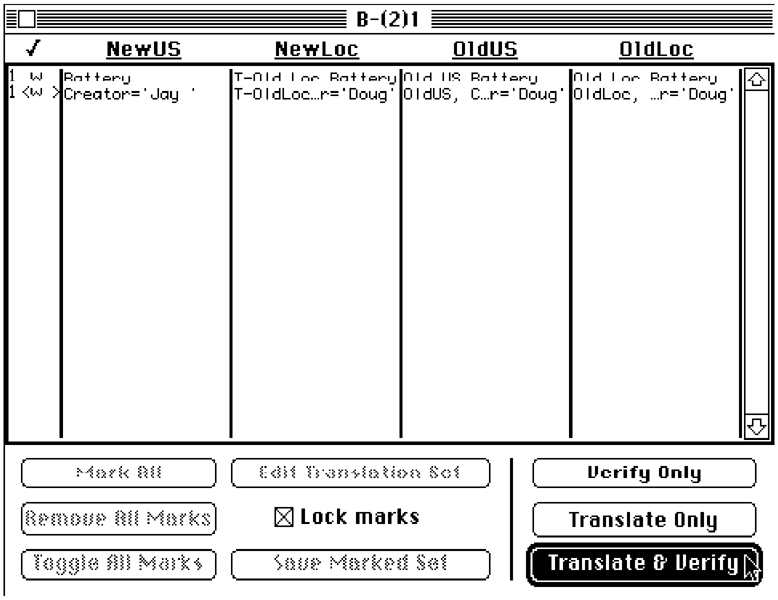
By the way, the creator types of the Work Glossary, Application Database, and History files are stored in AppleGlot's resources. By default, the files receive a creator type of 'MPS ', the MPW Shell type. You may change them to the creator type of your favorite text editor.

The creator type applied to the Work Glossary and Application Database files may be found in 'STR ' resource, ID 10195. The creator type applied to the History file may be found in 'STR ' resource, ID 10194.

The Work Glossary files are somewhat different than those produced by earlier versions of AppleGlot. For instance, in addition to AppleGlot's own resource item description, you will now find *real* ResEdit item numbers. There are other changes as well, but you do not need to know all of them in order to translate the text content of the resources.

<u>STEP 6: UPDATE NEWLOC FILE</u>

To update the *NewLoc* file(s), return to AppleGlot, and select "Translate & Verify".



✓	NewUS	NewLoc	OldUS	OldLoc
1 w	Battery	T-Old Loc Battery	Old US Battery	Old Loc Battery
1 <w >	Creator='Jay '	T-OldLoc…r='Doug'	OldUS, C…r='Doug'	OldLoc, …r='Doug'

NOTE: The marks in the left hand column of the *Environment* window will now indicate the status of the files listed to the right. These status marks replace the check mark and their presence indicates that the line is still marked. The following items may appear in the √ column:

- The digit to the left will be either a 1 (indicating a *glossary cycle* has been completed) or a 2 (indicating that at least one *update cycle* has occurred).
- The word 'abort' indicates that processing was aborted either by the user or AppleGlot.
- The word 'error' indicates that AppleGlot experienced an error during processing
- A pair of "< >" brackets indicates that text is present.
- A "w" within the brackets indicates that at least one !!WARNING message is present in the history file.
- A "v" within the brackets indicates that the last cycle that occurred was a *verification cycle*.

STEP 7: PRODUCE FINAL LOCALIZED SOFTWARE

Repeat **STEP 5** and **STEP 6** as many times as are needed to produce your final, translated piece(s) of software.

Appendix A Facts and New Features in AppleGlot v2.0

AppleGlot 2.0 is System 7 dependent. It won't work under older system software and it hasn't been tested on older Macintosh computers yet. Problems may be encountered on 68000 CPUs.

Comments within { } may be placed in the Work Glossary file.

See Appendix C.

True ResEdit numbers are in the Work Glossary to help in locating items when using ResEdit

See Appendix C.

Variant TMPLs based on Resorcerer field types

Only partially implemented - not all types supported yet. AppleGlot 2.0 currently supports only those fields we needed for our own projects. Time permitting we plan on implementing the rest of the types.

Background operation - switch to background under MultiFinder.

AppleGlot 2.0 isn't the CPU pig that the 1.0 family of AppleGlot is. You can get some work done while AppleGlot is chugging away in the background. AppleGlot is faster when in the foreground because it gives up time when in the background so that you can download your AppleLinks or whatever. Some programs (games in particular) don't give any CPU time to the background so AppleGlot won't get any work done in this case. Other applications (such as background compression programs) may come to life and stop AppleGlot from processing until they finish. We haven't tested AppleGlot while screen savers are active. AppleGlot may be suspended by some screen savers.

WARNING! AppleGlot can't protect itself from being set to work on a batch of files and then having those files moved by the user while AppleGlot is in the background. Its behavior in these cases is basically untested at this time.

Batch mode operation

All translations are handled in batches. Even a single file translation is a batch of 1.

TMPL and Language Glossary preferences

Standard default TMPLs are built in to AppleGlot. If you have several projects which needs a special TMPL definition to override these built in definitions, you may place it in a folder established via AppleGlot's preferences dialog and it will take precedence. TMPLs specific to a single project may be placed in the *Environment's* _TMPL folder and it will take precedence over all others.

Similarly, if you have several projects which needs a special Language Glossary, you may also place it in a folder established via AppleGlot's preferences dialog and it will take precedence. Language Glossaries specific to a single project may be placed in the *Environment's* _LG folder and it will take precedence over all others.

New logic provides a 'Wild Guess' based on item positions in the resource when there is no exact text match

Note: The guess may be useful or it may be totally wrong - AppleGlot can't tell but it will provide the text for you to examine in the Work Glossary. If it is close to being correct then you may make some small changes and save lots of time. If it makes no sense at all, you simply delete it and provide the correct translation as before. See Appendix C.

Folder based environment

See Appendix B.

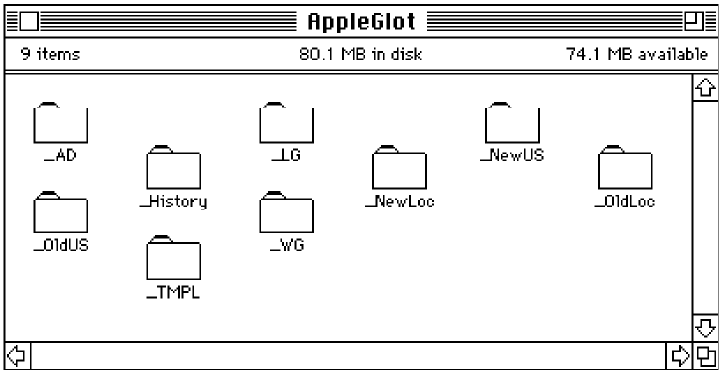
Automatic association of files within the environment based on type and creator

User may override or force associations where AppleGlot is unable to decide correctly.

Retrys individual resources without the speed zone if the speed zone wasn't large enough.

Appendix B The *Environment*

An AppleGlot 2.0 Environment consists of an *Environment* folder which contains nine folders, as shown below.



An environment is created for each user selected batch. **DO NOT** attempt to reuse old environments! The following folders contain text files that can be created or edited by any text processor that supports a “Save As Text” option. ‘MPS’ is the default creator type and ‘TEXT’ is the file type.

_AD Contains the Application Database file(s), a list of all the strings that AppleGlot found.

_History Contains the History file(s), the detailed information on all the resources in the NewUS/NewLoc file(s). Also contains the Batch History File, a summary of all warnings that have occured within the *Environment*. History files are described in detail in Appendix D.

_LG Contains the Language Glossary file(s), collection(s) of common strings and their translations.

_WG Contains the Work Glossary file(s), collection(s) of string for which AppleGlot found no precise translation. Work Glossaries are described in detail in Appendix C.

The following folder contains Resorcerer/ResEdit-style templates used for extracting text from resources.

_TMPL Contains the TMPL file(s), file(s) who’s resource forks contain one or more ‘TMPL’ resources. These TMPLs will override TMPLs contained in your designated Template preferences folder, as specified in the “Preferences...” dialog, and will also override any templates found within AppleGlot itself. If a particular TMPL appears twice, the last one found will be used!

The following folders contain the software being localized in its various version or new/old and U.S./localized versions.

_NewLoc Contains the new localized software file(s), the localized version we are currently creating.

_OldLoc Contains the old localized software file(s), the previous localized version.

_NewUS Contains the new U.S. software file(s), the latest U.S. version.

_OldUS Contains the old U.S. software file(s), the previous U.S. version that was used to create the“OldLoc” file(s), above.

Customizing the Environment

The names of the folders may be changed to suit the user’s preference. The names may be found in AppleGlot’s ‘STR#’ resource, ID 10005. You may edit the names with ResEdit, Resorcerer, or any other resource editor. Please note that there are two occurrences of the name “_History”. Both may be used. To avoid confustion, it is recommended that any change to one should also be applied to the other.

Appendix C The Work Glossary File

The Work Glossary file is a collection of strings for which AppleGlot found no precise translation. It identifies the source of the string—the resource type, ID, and name. It is also where you must fill in your translations.

Work Glossary Files may begin with a comment enclosed in “{ }” brackets. Additionally, each entry may also be preceeded with a comment as in the example below.

```
{This translated text came from your Language Glossary.}
DITL 428 dialog item text 24 (4) [?? Language Glossary Guess]
        <apple>
        <pomme>
```

In this example, the first line is a comment.

The second line identifies the resource:

DITL	The resource type
428	The resource ID
dialog item text	The template's field name
24	AppleGlot's template expansion number
(4)	These are the template expansion numbers as ResEdit will display it. Typically you will see numbers such as (1) or (23) but it is possible that you will encounter number sets such as (1,5) which occurs as nested loops are expanded.
[?? Language Glossary Guess]	AppleGlot's comments regarding the contents of the localized text (line four, below).

The third line displays the *NewUS* string, the string to be translated, enclosed in "< >" brackets.

<apple>	The NewUS string

The fourth line displays the *NewLoc* string,the string that has been translated, enclosed in "< >" brackets. The brackets will be empty if no guess is supplied.

<pomme>	The NewLoc string

NOTES: The AppleGlot parser is <u>not</u> equipped to handle comments at the end of a file.

Don't put comments on or between the "< >" brackets because the parser accepts everything between these delimiters as literal—comments will be considered as part of the literal.

<u>Appendix C</u> <u>The Work Glossary File (cont.)</u>

```
{You may place comments at the start of the work glossary file, but not at the end.}



{You may also place comments above any context line.}
DITL 128 dialog item text 30 (5) {Placing comments at the end of context lines is ok too.}
        <Extra NewUS Radio Button>
        <>

{This translated text came from your Language Glossary.}
DITL 428 dialog item text 24 (4) [?? Language Glossary Guess]
        <Extra NewUS Check Box>
        <NewUS Check Box>



{The translation supplied here was found within the Application Database.}
DITL 328 dialog item text 24 (4) [?? Application Database Guess]
        <NewUS radio button text>
        <OldUS NewUS Check Box>




{This type of translation comes from the Work Glossary and should only happen if you aborted
AppleGlot and restarted with a partially completed Work Glossary.}
DITL 228 dialog item text 24 (4) [?? Work Glossary Guess]
        <Extra NewUS Check Box>
        <Extra NewUS Check Box text>

{During the start of the first update cycle AppleGlot reads all the strings from the OldUS and
OldLoc files and puts the strings into an internal structure to use as best guesses when an exact
context translation can't be found.}
DITL 128 dialog item text 24 (4) [?? Internal Guess]
          <Extra NewUS Check Box>
          <Extra OldLoc Check Box>


{When all else fails AppleGlot attempts to provide you with the text from the same position in the OldLoc resource. It may be
almost correct or it may be totally wrong.}
DITL 129 dialog item text 24 (4) [?? Wild Guess]
          <Extra NewUS Check Box>
          <extra newus check box>

DITL 529 dialog item text 24 (4) [?? Wild Guess]
          <Extra NewUS Check Box>
          <This text is totally crazy and you should replace it with the correct text.>

{The blank <> is what happens when AppleGlot can't find anything at all to use as a guess.}
DITL 128 dialog item text 30 (5)
          <Extra NewUS Radio Button>
          <>

{In this case the ResEdit number (n) is missing. This happens when the TMPL has no loops in it
thus ResEdit will not use numbers.}
DITL 128 text 1
          <Some Text>
          <>
```

Appendix D The History File and Batch History File

One Batch History File is created for each *Environment*. It contains warnings that were found in each of the individual History Files within the *Environment*. Thus, the Batch History File can be read for a quick overview of potential problems, if any.

History File and Batch History File Structure

The History File and Batch History File consists of three parts: the header, the body, and the footer; repeated for each cycle.

The header

The header identifies the cycle, filename of the New Localized software, starting time/date of this cycle, and filename and version of the AppleGlot that created this History File.

INTERNAL CYCLE: (Pass 0) AppleGlot creates an internal language glossary using the old US and Localized files. This allows full use of all matched text regardless of resource ID.

GLOSSARY CYCLE: (Pass 1) AppleGlot creates a work glossary and an application database. The application database contains the strings. The work glossary contains the string that require a manual translation. Some strings in the work glossary may already have a suggested translation, coming from the internal or external language glossary or from a wild guess.

UPDATE CYCLE: (Pass 2) When a user selects ‘Translate”, AppleGlot will perform an update of the New Localized file using the strings translated in the work glossary.

VERIFY CYCLE: AppleGlot verifies the consistency of the New US and Localized application and reports into the History File.

The body

The body in a History File begins with a line which identifies the fields within the records:

TWA Type ID Resource Name USSize->NewLoc SysPrgLocProPreExt Action codes

- **TWA** - This field contains three Boolean flags:

a the template for this resource was built in to AppleGlot
l the template for this resource was found in the environment's _TMPL folder
p the template for this resource was in the user specified preferred templates folder
w this resource has text which has been written into the work glossary
a this resource has text that has been updated into the application database

- **Type** - The resource type
- **ID** - The resource ID
- **Resource Name** - The resource name (truncated, if necessary, to make it fit)
- **USSize->NewLoc** - The size of the resource in the US version and the size of the resource in the new localized version.
- **SysPrgLocProPreExt** - Six Boolean flags indicating the resource's attributes ("..." indicates that the attribute is not set):

Sys - read into system heap
Prg - purgeable
Loc - locked
Pro - protected
Pre - preload
Ext - extended resource attributes

* * * * * * * * * * * * * * * * *
* See the Resource Manager *
* chapter in <u>Inside Macintosh</u> *
* for a detailed discussion of *
* resource attributes. *
* * * * * * * * * * * * * * * * *

- **Action Codes** - indicates the actions that have been performed by AppleGlot on this resource (action codes are defined below)

The body contains records for each resource. Each record contains a number of fields containing useful information about the translation of each resource. Some records may contain "??WARNING" or "??FATAL" entries. These are the only entries that appear in the body of records that are in the Batch History File. (Warning and Fatal messages are described in greater detail below.)

<u>Appendix D</u> <u>The History File and Batch History File (cont.)</u>

The Footer

The footer indicates that it is the end of the processing of the named New Localized software processing and shows the end time/date of this cycle

Action Codes

1 The resource translated successfully. There is a template.
2 The New-Loc resource did not yet exist. There is a template.
3 Copied some items from New-Loc resource. There is a template.
4 Preserved some items from Old-Loc resource. There is a template.
5 Preserved some items from New-Loc resource. There is a template.
6 There is no New-Loc resource yet. The Old-US resource matches the New-US so the Old-Loc was copied. There is a template.
7 There already is a New-Loc resource so it was not changed. No template.
8 *{Not currently used.}*
9 *{Not currently used.}*
10 The Old-US matches the New-US so the Old-Loc was copied. No template.

11 *{Not currently used.}*

12 New-loc resource already exists and was not modified. No template.

13 New-US resource copied. No template.

14 Copied Extra resource from Old-Loc to New-Loc.

15 Extra resource in Old-Loc NOT copied.

16 Previous translations strings preserved by inserting them into the internal Language Glossary prior to translation.

17 Verified resource. There is a template.

18 Verified resource. No template.

19 Extra resource.

20 Count error.

21 Exact match located in Work Glossary.

22 Inexact match located in Work Glossary.

23 Exact match located in Application Database.

24 Inexact match located in Application Database.

25 Exact match located in Language Glossary.

26 Inexact match located in Language Glossary.

27 Wild guess from Language Glossary using context only. It may be close or totally wrong.

Appendix D The History File and Batch History File (cont.)

??WARNING and ??FATAL codes

??WARNING #1 If the history file pre-allocation fails we get this message. In order to be sure that there is at least a reasonable amount of disk space AppleGlot pre-allocates some space for the history file. It may be allocating more space than is actually necessary. The STR resource named kMinimumDiskSpaceWarningSizeStr contains a number which is the size that AppleGlot uses when allocating the history file.

#2 *{Not currently used.}*

??WARNING #3 When packing a resource after expanding and translating it if AppleGlot finds that the size of the data for an AUTO field isn't 2 bytes it outputs this message. This shouldn't be possible and is most likely the result of an internal AppleGlot bug.

??WARNING #4 If the data forks of both the New-US and New-Loc files are the same length AppleGlot will check them byte-by-byte and if any single byte is different it will report the problem.

??WARNING #5 During translation and verification AppleGlot needs to load each resource into memory to work on it. Prior to each load AppleGlot gets the size of the resource and asks the memory manager if there is enough room for the resource. If the memory manager say no we get this error message. When doing a translation pass and there is a previous translation to examine for reusable items AppleGlot needs to load three copies of each resource into its heap. When it is done working on them it creates a fourth copy and puts it into the New-Loc file. Usually translatable resources aren't too gigantic so they will fit without trouble unless you try to run AppleGlot in a much smaller space then it likes to run in. Non-translatable resources can be a bit of a problem sometimes. AppleGlot still needs to load copies into memory to examine them byte-by-byte to see if the New-US and Old-US resources are identical so it can copy the Old-Loc resource into the New-Loc file. Sometimes application files contain non-translatable resources greater than 600K in size and if you try to get four of them into the heap at the same time AppleGlot isn't going to like it very much and will tell you that it failed to work correctly. The only way to fix this situation is to increase the amount of memory allocated to AppleGlot and try again.

??WARNING #6 During translation pass #1 the Work Glossary text file is created and each text item which AppleGlot can not find an exact translation for will be written to this file. After translation pass #1 has completed the Work Glossary has a complete list of everything that AppleGlot couldn't figure out. During translation pass #2 every possible translation is assumed to be accounted for in either the Work Glossary or the Application Database and no more errors need to be written into the Work Glossary. If anything goes wrong and AppleGlot encounters something it didn't expect to encounter in translation pass #2 this message reports that it had a problem. It often means that something in the Work Glossary context line or <New-US text> line was accidentally changed resulting in a translation changing its identity and confusing AppleGlot into sending a new unknown string into the Work Glossary. The original Work Glossary is renamed and your efforts at translation are not lost but you should examine the older Work Glossary or Application Database to see if there is something wrong which could be corrected. One thing which could cause this to happen is if you translate pass #1 without an external templates file and then use an external templates file the second time you translate. In effect you've given AppleGlot more information the second time then it had the first time and it found more strings.

??WARNING #7 This would most likely be the result of an internal AppleGlot bug. During verification AppleGlot gets all resources of a given type in map index order. It gets a count of the number of resources of a given type in the New-US file and then proceeds to get each of them in turn. If the resource somehow fails to load AppleGlot reports that it couldn't get the resource which it was told it could have.

??WARNING #8 When AppleGlot is verifying the New-US and New-Loc files it examines the attribute flags of both files. If they are not identical it reports the problem. This usually means that someone has used ResEdit to modify the attributes of the New-Loc file.

??WARNING #9 AppleGlot compares the resource map attribute of the New-US and the New-Loc files. If they are not identical this message reports the error. This usually means that someone has used ResEdit to modify the attributes of the New-Loc file.

??WARNING #10 During verification AppleGlot gets all resources in the New-US file of a given type in map index order. It expects every resource in the New-US file to be in the New-Loc file as well. If it fails to get a resource from the New-Loc file when it requests it by type and ID AppleGlot will report this message. You could get this message if you used ResEdit to add an extra resource to the New-US file after it was translated and then verified the two files. You would be more likely to get this message due to deleting a resource from the New-Loc file after translation but prior to verifying the two files.

??WARNING #11 This situation is usually quite normal but AppleGlot reports it anyway. During verification there is a special sweep of the New-Loc file looking for resources which aren't in the New-US file. If an extra resource is found in the New-

Loc file which isn't in the New-US file this message reports the news. This means that there may have been resources manually moved into the New-Loc file by ResEdit or there may have been extra resources in the Old-Loc file which were copied into the New-Loc file during translation.

??WARNING #12 When a resource without a TMPL in the New-Loc file is found to be of the same size as the same resource in the New-US file AppleGlot expects the internal bytes to be identical as well. If so much as one byte is different then AppleGlot writes this comment. It is possible that there was one or more small modifications made with ResEdit which AppleGlot doesn't know anything about. If AppleGlot copies a resource from the Old-Loc file into the New-Loc file it is possible that the differences were made by ResEdit during an earlier localization.

??WARNING #13 If there is no TMPL for a resource then AppleGlot won't change it and expects it not to change. When AppleGlot finds a difference in the sizes of the New-Loc and New-US resources it writes this comment. There are times that these resources will be changed and not be the same size such as when a PICT resource is changed during localization. If a changed PICT (or other type) resource is copied from the Old-Loc to the New-Loc files AppleGlot will still report that the resource size is not the same as the resource in the New-US file. It is up to the user to check that the resource in the New-Loc file is valid. Note that the Signature and 'vers' resource types are special and won't be checked even though they have no TMPL in AppleGlot.

??WARNING #14 If the verify pass finds that the name of the New-Loc resource isn't identical to the name of the New-US resource then AppleGlot writes this message into the history file. It may or may not be of any importance that the names match. It depends on whether or not the resource is called from the program by name or by ID. If it is called by number then the name is of no concern but if the resource is called by name then the program will fail to find the resource.

??WARNING #15 This one means that the user clicked on the CANCEL button in the progress dialog during the verification pass and the verification was aborted before it was completed.

Appendix D The History File and Batch History File (cont.)

??WARNING #16 During verification the New-US and New-Loc resources are expanded and inspected on a field by field basis. If any of the fields which are not RECTs and are not text are not identical this message informs us that some item has changed which AppleGlot didn't expect to change. This will happen if you use ResEdit to modify some numeric field or something similar and then run AppleGlot again. Since you changed a field that AppleGlot would not have changed it will report that there is a problem. You will need to use ResEdit to examine the two resources to see what has changed and if it is something you find acceptable or not. It is quite possible that you didn't run ResEdit on the resource but instead AppleGlot copied it from the Old-Loc file for you. In this case the result is still the same, a non-translatable field is different in the New-Loc resource and AppleGlot will write this comment into the history file.

??WARNING #17 Fields of type CSTR are expected to have a terminating NULL character. If during expansion of the resource the end of the block is reached without encountering a NULL this is the message which informs you of the problem.

??WARNING #18 If during the expansion of a resource the template indicated that the resource has been fully expanded but there remains more data in the resource block beyond AppleGlots byte pointer this message will be written. Either the template is incorrect or the resource truly has some extra bytes after all the valid stuff. I suppose this could happen if AppleGlot gets lost due to an internal bug.

??WARNING #19 If during expansion of a resource item of type PSTR the data should become larger than 255 bytes this message will be written into the history file. Since the PSTR field takes the first byte of the data as its length and a byte can't exceed FF(hex) or 255(decimal) this string could only occur if there was a bug in AppleGlot itself which caused the block containing the PString characters to expand beyond 255 bytes.

??WARNING #20 When a translated PSTR, ESTR or OSTR field exceeds 255 characters in length you will get this message because a PString can only hold 255 characters at most. The resulting resource will have the string in this field truncated to 255 characters so it will be a valid resource but its contents in the particular field in question will be incomplete.

#21 *{Not currently used.}*

??WARNING #22 When expanding a resource field of type P240 the maximum number of characters in the PString is 240. If the length byte of the PString is greater than F0(hex) or 240(decimal) then this message is written into the history file. P240 is a special field type designed to check DITL text items for correctness. The DITL will work correctly but Inside Mac states that text items have a maximum of 240 characters and when a DITL with more than 240 characters is opened in ResEdit the editor will usually complain that some field following the text item is incorrect. Editing with the ResEdit TMPL editor will work in spite of the invalid field.

??WARNING #23 When AppleGlot encounters an extra resource in the Old-Loc file which wasn't in the New-US file and the user chooses to copy it into the New-Loc file AppleGlot will write this line into the history file.

??WARNING #24 When, during translation, AppleGlot encounters extra resources in the Old-Loc file but the user chooses to not copy them into the New-Loc file AppleGlot will always make a note of the fact that it is skipping the resource so you will later know exactly what it did when you read the history file.

??WARNING #25 AppleGlot isn't prepared at the present time to deal with ANY of the resource extensions. The only ones defined so far are compression schemes. This message will occur during verify when the resource attributes are checked and the Extended bit is set. During translation this message may appear when the New-US resource is loaded for translation if the Extended bit is set. It can also occur when an extra resource is copied from the Old-Loc file into the New-Loc file. In addition it may happen if the New-Loc resource has had its Extended resource bit set somehow. There have been cases where the Extended bit was set in a resource but ResEdit didn't put a check in the Compressed item box (I suppose that the resource wasn't actually compressed). When the resource was DeRez'd it was found to have attributes of $01 so be sure you use DeRez to check it as well as using ResEdit.

??WARNING #26 During verification there is a special sweep of the New-US file looking for resources which aren't in the New-Loc file. If extra resource are found in the New-US file which aren't in the New-Loc file this message reports it. Usually this means that there have been resources manually removed from the New-Loc file by ResEdit.

??WARNING #27 When a resource is being expanded and the template tells AppleGlot to expect more data but the resource suddenly ends we post this message into the history file. Generally AppleGlot will continue to parse the template and fill the expansion fields with NULL characters to pad in the missing pieces. The resulting resource will conform to the expected size but its fields may not contain the correct data if NULL isn't valid.

??WARNING #28 When expanding a 'P0nn' field in a resource this message will appear in the history file if the length byte of the PString is greater than nn. If the PString was created using a hex editor in ResEdit this could occur if you don't pay attention to the constraints of the P0nn maximum size.

??WARNING #29 This string appears if the WorkGlossary (or Application Database) contain a translated string which is too long to fit the specified size for a P0nn field. nn is two hex characters which specify the maximum length of the PString. For example, if the field is P031 and the translation is longer than 31(hex) or 49(decimal) characters you'll get this error message.

??WARNING #30 When expanding Cnnn fields AppleGlot expects to find a NULL character someplace within the limits of the field. If no NULL is located this is the message telling us that something is wrong with the CString assigned to the field.

??WARNING #31 During expansion of LSTZ fields in a resource AppleGlot expects the field to be terminated by a NULL byte. LSTZ fields are the last group of fields in a resource and if AppleGlot should happen to encounter the end of a resource without finding the termination NULL it will put this message into the history file. See the TMPL for MENU for an example of a LSTZ loop.

??WARNING #32 When AppleGlot is moving a translated resource into the New-Loc file it must put it into the same resource map position so the resource map sequence remains unchanged. In order to do this AppleGlot keeps the handle to the resource which it pulled out of the New-Loc file which it used to preserve any non-translatable items. It then tosses away the contents of the handle and substitutes the newly packed resource for the original New-Loc resource. It next makes a call to ChangedResource to tell the resource manager that it needs to rewrite the resource into the New-Loc file using the old handle. If there is anything wrong in AppleGlots internal logic and the call to ChangedResource fails AppleGlot will write this message into the history file.

??WARNING #33 This one may be more annoying then helpful. If a CString is shorter than the full 'Cnnn' field size this message is placed into the history file. It seems that the rules for 'Cnnn' fields allow for CStrings from zero length up to nnn-1 in length.

Appendix D **The History File and Batch History File (cont.)**

??WARNING #34 This could happen during packing of a resource if the text assigned into a field of type 'Cnnn' is too large to fit the field. 'Cnnn' is a variable sized CString which isn't allowed to exceed nnn in length where nnn is three hex characters. When this message appears in the history file AppleGlot has truncated the CString by simply placing a NULL in the last position in the field.

??WARNING #35 During expansion of resources AppleGlot is using the field type from the template resource to tell it how to expand each field in the resource. If there is a template field which is new that AppleGlot doesn't yet know about it will output this string into the history file. If there is a simple misspelling in the template type field the result will be the same.

??WARNING #36 When AppleGlot is packing a resource it has just expanded and finds an expansion type that it doesn't recognize this is the string that is displayed. It is the result of an internal logic error in AppleGlot itself. If it understood what the type was during expansion it should also understand the same type during packing. Since the expansion and packing routines are entirely separate this is basically an error message used to keep the programmer on his toes.

??WARNING #37 During verification AppleGlot expands resource pairs from both the New-Us file and the New-Loc file. It keeps a block in memory of all the expanded pieces as well as the labels from the template resource it used to do the expansions. When doing the compares there is a consistency check to see that AppleGlot itself hasn't somehow lost track of the expanded pieces. The only way this message should ever come up is if AppleGlot itself makes a mistake.

??WARNING #38 When AppleGlot does its compares of expanded resource pairs one of its parameters to the routine which does the compares controls whether or not it checks fields which are defined as text. This message would only occur if AppleGlot

??WARNING #39 This happens if the user clicks the CANCEL button in the progress dialog or if the history file itself isn't correctly initialized. But since the history file couldn't receive the string if it wasn't open this really only happens if you cancel the translation by clicking the button in the progress dialog.

??WARNING #40 This seldom seen string is the result of an empty flag pattern which would mean that AppleGlot didn't do anything to the resource being translated. It should only be seen if there is an internal AppleGlot logic error.

#41 *{Not currently used.}*

#42 *{Not currently used.}*

??WARNING #43 During the verification of resources AppleGlot compares the expanded resources on a field by field bases. It uses the descriptions in the template templates to step through the fields. This message is reporting that a field which is normally considered to be text and therefore variable is being treated as text which shouldn't change. In general this message shouldn't appear and would most likely be the result of an internal AppleGlot bug.

??WARNING #44 AppleGlot normally considers rects to be a translatable field which the user will most likely have modified using AppleGlots own rect editor or ResEdit. If for some reason this type of field is being treated as something which shouldn't have changed this message will appear. This would be considered an AppleGlot bug.

??WARNING #45 If while dealing with the various resources AppleGlot should happen to get out of sync with the various expansion blocks which contain the various components of each resource this message will warn that there has been a problem. It indicates that AppleGlot got lost and is most likely a bug.

Appendix D The History File and Batch History File (cont.)

??WARNING #46 When comparing expanded resources AppleGlot knows how many pieces each resource was divided into. If the number of pieces of each resource differs AppleGlot informs us with this message. This can occur when the number of items is changed during normal localization as well as when AppleGlot compares a bad New-US resource to a valid New-Loc resource due to the problems in parsing the bad resource.

??WARNING #47 During the expansion of a resource item of type PSTR the length byte indicated that the string contained more characters than there were bytes remaining in the block of memory containing the resource. AppleGlot detected the problem, truncated the string and used only those bytes which were within the block. The problem here is usually an incorrect length byte in the resource. It is also possible that the resource was shortened using the ResEdit hex editor which results in an unchanged length byte which is no longer in sync with the true length of the string.

??WARNING #48 During the expansion of a resource item of type WSTR the length word indicated that the string contained more characters than there were bytes remaining in the block of memory containing the resource. AppleGlot detected the problem, truncated the string and used only those bytes which were within the block. The problem here is usually an incorrect length word in the resource. It is also possible that the resource was shortened using the ResEdit hex editor which results in an unchanged length word which is no longer in sync with the true length of the string.

??WARNING #49 During the expansion of a resource item of type LSTR the length longint indicated that the string contained more characters than there were bytes remaining in the block of memory containing the resource. AppleGlot detected the problem, truncated the string and used only those bytes which were within the block. The problem here is usually an incorrect length longint in the resource. It is also possible that the resource was shortened using the ResEdit hex editor which results in an unchanged length longint which is no longer in sync with the true length of the string.

??WARNING #50 During the expansion of a resource item of type ESTR the length byte indicated that the string contains more characters than there are bytes remaining in the block of memory containing the resource. AppleGlot detected the problem, truncated the string and used only those bytes which were within the block. The problem here is usually an incorrect length byte in the resource. It is also possible that the resource was shortened using the ResEdit hex editor which results in an unchanged length byte which is no longer in sync with the true length of the string.

??WARNING #51 During the expansion of a resource item of type P240 the length byte indicated that the string contains more characters than there are bytes remaining in the block of memory containing the resource. AppleGlot detected the problem, truncated the string and used only those bytes which were within the block. The problem here is usually an incorrect length byte in the resource. It is also possible that the resource was shortened using the ResEdit hex editor which results in an unchanged length byte which is no longer in sync with the true length of the string.

??WARNING #52 During the expansion of a resource item of type OSTR the length byte indicated that the string contains more characters than there are bytes remaining in the block of memory containing the resource. AppleGlot detected the problem, truncated the string and used only those bytes which were within the block. The problem here is usually an incorrect length byte in the resource. It is also possible that the resource was shortened using the ResEdit hex editor which results in an unchanged length byte which is no longer in sync with the true length of the string.

Appendix D **The History File and Batch History File (cont.)**

??WARNING #53 During the expansion of a resource item of type Hnnn the length longint nnn indicated that the string contains more characters than there are bytes remaining in the block of memory containing the resource. AppleGlot detected the problem, truncated the string and used only those bytes which were within the block. The problem here is that the resource just isn't long enough to satisfy the nnn part of the Hnnn template field value. The template field may be wrong or the resource may be damaged.

??WARNING #54 During the expansion of a resource item of type P0nn the length byte nn indicated that the string contains more characters than there are bytes remaining in the block of memory containing the resource. AppleGlot detected the problem, truncated the string and used only those bytes which were within the block. The problem here is that the resource just isn't long enough to satisfy the nn part of the P0nn template field value. The template field may be wrong or the resource may be damaged.

??WARNING #55 During the expansion of a resource item of type Cnnn the length longint nnn indicated that the string contains more characters than there are bytes remaining in the block of memory containing the resource. AppleGlot detected the problem, truncated the string and used only those bytes which were within the block. The problem here is that the resource just isn't long enough to satisfy the nnn part of the Cnnn template field value. The template field may be wrong or the resource may be damaged.

??WARNING #56 AppleGlot was requested to allocate a larger relocatable block of memory then there was room for in the heap. This could be caused during resource expansion by incorrect length values for LSTR or WSTR or similar problems. It could also be caused by a bad template field descriptor values if memory is already tight. It may simply be that you allocated too little memory to AppleGlot when you launched it. Check the value in the 'Get Info' menu in the Finder to see if it's large enough.

??WARNING #57 During the expansion of a resource item AppleGlot couldn't allocate memory for an expansion of one of a resources fields. This gives you a bit of information about what field was being expanded when the problem was encountered. It may be a bad length longint for an LSTR or something similar or it may be that the field preceding this one was bad and caused this field to be out of place within the resource resulting in random junk being used as the items size.

??WARNING #58 This one is usually caused by a CHAR field accidentally being translated into more than one character. Check you Work Glossary or Application Database to see that there is only a single character in the translation.

??FATAL #59 This message means what it says. If AppleGlot is able to create its clone files and history file and then runs out of disk space when doing the translation it tries to put this message into the history file. It is possible that there isn't any room in the history file too and the dialog on the screen may be the only indication of what happened.

??FATAL #60 This can occur when a resource is too big to be loaded into memory. AppleGlot detects that it is too big to fit and logs the error into the history file. Once AppleGlot has failed to copy or translate a resource the NewLoc file is not complete and this is considered a fatal error. Usually the solution to this problem is to increase the heap space requested by AppleGlot during launch. Another possible fix would be to disable the Speed Zone logic, thus freeing all of AppleGlots internal heap space for general use. This would cause a much slower translation but that may be better than not being able to translate at all.

??WARNING #61 An LSTR or WSTR length field had its high bit set.

??WARNING #62 During the expansion of a resource item of type LSTR the length longint indicated that the string contained less than zero characters. The problem here is usually an incorrect length longint in the resource or we got out of sequence due to a problem in the previous resource.

??WARNING #63 During the expansion of a resource item of type WSTR the length integer indicated that the string contained less than zero characters. The problem here is usually an incorrect length word in the resource or we got out of sequence due to a problem in the previous resource.

??WARNING #64 During the expansion of a resource item of type WSTR the length integer indicated that the string contained less than zero characters. The problem here is usually an incorrect length word in the resource or we got out of sequence due to a problem in the previous resource.

??WARNING #65 The number of bytes in the data forks of the New-US and the New-Loc files was different during the verification of the two files.

??WARNING #66 During expansion of template resource AppleGlot a field type it doesn't know how to handle. You may have misspelled the four character name or you may be using a field type that is valid in ResEdit or Resorcerer but that has not yet been implemented in AppleGlot.

??WARNING #67 The ZCNT loop count indicates that we should find more data to continue at least one more cycle but we ran out of data in the resource.

??WARNING #68 The OCNT loop count indicates that we should find more data to continue at least one more cycle but we ran out of data in the resource.

??WARNING #69 The expansion of the resource failed before it could start. I've seen this happen on resources which contained no data at all which causes the template expansion logic to abort before it really gets to start.

??WARNING #74 There was a resource in the OldLoc file which didn't exist in the OldUS file. When AppleGlot tried to copy it into the NewLoc file it found there was already a resource with the same type and I.D. in the NewLoc file. This can happen if the localizer added an extra resource to the OldLoc file and the NewUS file happened to use the same type and I.D. for a new resource. It is up to the localizer to examine the resources and see if they wish to preserve the OldLoc extra resource by renaming or renumbering it.

??WARNING #75 An expansion field contained data indicating that the field contained more data then there was actual bytes remaining in the resource block. Such a problem could come from a PSTR field where the length byte was incorrect and indicated that there were more characters then the block held. The NewLoc field is truncated to the number of bytes that actually are available in the block to avoid copying random data from beyond the end of the resource. Such random data would more than likely cause the Work Glossary to be rewritten on a future translation pass when the random data couldn't be matched.

??WARNING #76 Some routine encountered some problem and cleared the sCycleNormal flag. This global flag controls the individual cycles within the overall batch process. The routine which encountered the problem also fills in the ^0 parameter from the kCycleMessageStrings list of message strings.

??WARNING #77 Something is wrong with a glossary file. Typically what has happened is that the person doing the localization has accidentally changed a context line or messed up the <NewUS> or <NewLoc> text bracketing somehow. Perhaps there are invisible characters outside of the <TEXT> group which can't be seen without turning on the 'Show Invisibles' command of your editor. In any case AppleGlot has stumbled on parsing the file and will put the context lines from the previous translation as well as from the translation it couldn't read into the history file for the user to examine and for them to copy to use to search in the problem glossary to at least get them close to the problem so that finding the trouble isn't usually too hard. In the case where the problem is with the very first translation there will be no previous context, only a blank line.

#78 When we finish translation we take one final look to see if there is a resource with the same type as the CREATOR for the file and with and ID of 0. If there is we assume it is the signature resource and translate it as a 'STR ' resource. It will therefore be listed twice in the history file - once as a resource without a template and once as the signature. AppleGlot will only try to translate it as a 'STR ' if the length byte of the NewUS resource is correct for a PSTR which takes up the total number of bytes in the resource block.

??WARNING #79 This means we ran out of room in the speed zone so either turn it off or allocate more size. In addition to the above options you may elect to modify the kSpeedZoneMaxExpansions resource to increase its value to allocate more room all the time. The general situation is that AppleGlot creates an internal zone called the Speed Zone in which it does its resource expansion. If this zone is not sufficiently large to hold all the expansions of all the resources then the NewHandle call will fail and return a memFullErr. When this happens AppleGlot is stuck because it no longer has enough memory to do its job and the resource is incorrectly expanded. It used to go on and pack the bad expansion and create a bad resource but instead it now detects this problem and destroys the speed zone and does a retry. This message is output prior to that retry so if the retry works ok or if it fails there will be another message to explain that success or failure. AppleGlot reports that it had a problem so the user is aware of the situation and can pay special attention to the translated resource to be sure all is truly ok. If the user chooses the option of turning off the Speed Zone entirely this will give AppleGlot its entire heap for expansions but may result in very slow translations of large resources.

??WARNING #80 This means we ran out of room in AppleGlots main heap zone. The Speed Zone wasn't involved. The only chance for improvement here is to allocate more memory by using the Finders 'Get Info' menu and adjusting the memory requested.

??WARNING #81 This message indicates that there was not enough memory while expanding a resource within AppleGlots heap. Normally the Speed Zone is used for expansions but if the first try fails the Speed Zone is eliminated and a second try is attempted within AppleGlots heap. Due to the large size of the Speed Zone the free space within AppleGlots heap is usually sufficient for the second try at expanding the resource. In the case where there was no Speed Zone to delete or when the resource is just too huge for AppleGlot to handle in the currently allocated memory the expansion will fail.

??WARNING #82 When the Speed Zone size isn't sufficient to translate the various resources AppleGlot will delete it and try the translation again in AppleGlots normal heap. If this second try fails AppleGlot will put this message in the history file to inform the user that the translation didn't succeed.

Appendix D **The History File and Batch History File (cont.)**

?? #83 When the Speed Zone size isn't sufficient to translate the various resources AppleGlot will delete it and try the translation again in AppleGlots normal heap. If this second try is successful AppleGlot will put this message in the history file to inform the user that the translation was eventually completed without further problem.

Appendix D **The History File and Batch History File (cont.)**

Sample History File

```
INTERNAL CYCLE - 'T-Battery.OldLoc' begin at 10:17:59 AM 9/29/92 - 'AppleGlot 2.0'
TWA Type      ID        Resource Name       USSize->NewLoc SysPurLocProPreExt  Action codes
a  'DITL'    -15616  "                "     136     136    ...Prg...........   :16:
a  'DITL'    -15614  "                "     122     122    ...Prg...........   :16:
a  'ALRT'    -15614  "                "      12      12    ...Prg...........   :
a  'ALRT'    -15613  "                "      12      12    ...Prg...........   :
a  'STR#'    -15616  "                "     552     552    ...Prg...........   :16:
a  'DLOG'    -15616  "                "      22      22    ...Prg...........   :16:
End of 'T-Battery.OldLoc' at 10:17:59 AM 9/29/92

GLOSSARY CYCLE - 'T-Battery.OldLoc' begin at 10:18:00 AM 9/29/92 - 'AppleGlot 2.0'
TWA Type      ID        Resource Name       USSize->NewLoc SysPurLocProPreExt  Action codes
a a'DITL'    -15616  "                "     136     136    ...Prg...........   :1:3:4:25:
a a'DITL'    -15614  "                "     122     122    ...Prg...........   :1:3:4:25:
a  'ALRT'    -15614  "                "      12      12    ...Prg...........   :1:4:
a  'ALRT'    -15613  "                "      12      12    ...Prg...........   :1:4:
a a'STR#'    -15616  "                "     552     552    ...Prg...........   :1:3:4:25:
a a'DLOG'    -15616  "                "      22      22    ...Prg...........   :1:3:4:25:
End of 'T-Battery.OldLoc' at 10:18:02 AM 9/29/92

VERIFY CYCLE - 'T-Battery.OldLoc' begin at 10:18:07 AM 9/29/92 - 'AppleGlot 2.0'
TWA Type      ID        Resource Name       USSize->NewLoc SysPurLocProPreExt  Action codes

  'DITL' -15616         ??WARNING #37 Non-translatable field mismatch. TMPL expansion #5. 'DBYT' {Item type}.
      NewUS  - HEX = '80'. Field length = 1. Mismatch at 0.
      NewLoc  - HEX = '00'. Field length = 1. Mismatch at 0.
  'DITL'     -15616  "                "     136     136    ...Prg...........   :17:

  'DITL'     -15614  "                "     122     122    ...Prg...........   :17:
  'ALRT'     -15614  "                "      12      12    ...Prg...........   :17:
  'ALRT'     -15613  "                "      12      12    ...Prg...........   :17:
  'STR#'     -15616  "                "     552     552    ...Prg...........   :17:
  'DLOG'     -15616  "                "      22      22    ...Prg...........   :17:
Map attributes match    . . . . . . . . . . . . . . . .
New-US and New-Loc have NO data fork
Data fork logical sizes      = 0  0
Data fork physical sizes     = 0  0
Resource fork logical sizes  = 1270  1270
Resource fork physical sizes = 1536  1536
End of 'T-Battery.OldLoc' at 10:18:08 AM 9/29/92

UPDATE CYCLE - 'T-Battery.OldLoc' begin at 10:18:35 AM 9/29/92 - 'AppleGlot 2.0'
TWA Type      ID        Resource Name       USSize->NewLoc SysPurLocProPreExt  Action codes
a a'DITL'    -15616  "                "     136     136    ...Prg...........   :1:5:23:
```

```
a a'DITL'      -15614  "              "     122     122    ...Prg...........      :1:5:23:
a  'ALRT'      -15614  "              "      12      12    ...Prg...........      :1:5:
a  'ALRT'      -15613  "              "      12      12    ...Prg...........      :1:5:
a a'STR#'      -15616  "              "     552     552    ...Prg...........      :1:5:23:
a a'DLOG'      -15616  "              "      22      22    ...Prg...........      :1:5:23:
End of 'T-Battery.OldLoc' at 10:18:37 AM 9/29/92

VERIFY CYCLE - 'T-Battery.OldLoc' begin at 10:18:42 AM 9/29/92 - 'AppleGlot 2.0'
TWA Type       ID          Resource Name         USSize->NewLoc SysPurLocProPreExt  Action codes

  'DITL' -15616            ??WARNING #37 Non-translatable field mismatch. TMPL expansion #5. 'DBYT' {Item type}.
        NewUS  - HEX = '80'. Field length = 1. Mismatch at 0.
        NewLoc  - HEX = '00'. Field length = 1. Mismatch at 0.
  'DITL'       -15616  "              "     136     136    ...Prg...........      :17:

  'DITL'       -15614  "              "     122     122    ...Prg...........      :17:
  'ALRT'       -15614  "              "      12      12    ...Prg...........      :17:
  'ALRT'       -15613  "              "      12      12    ...Prg...........      :17:
  'STR#'       -15616  "              "     552     552    ...Prg...........      :17:
  'DLOG'       -15616  "              "      22      22    ...Prg...........      :17:
Map attributes match    . . . . . . . . . . . . . . . .
New-US and New-Loc have NO data fork
Data fork logical sizes      = 0  0
Data fork physical sizes      = 0  0
Resource fork logical sizes  = 1270  1270
Resource fork physical sizes = 1536  1536
End of 'T-Battery.OldLoc' at 10:18:43 AM 9/29/92
```

Appendix D The History File and Batch History File (cont.)

Sample Batch History File

```
INTERNAL CYCLE - 'T-TeachText.OldLoc' begin at 10:17:48 AM 9/29/92 - 'AppleGlot 2.0'
End of 'T-TeachText.OldLoc' at 10:17:51 AM 9/29/92

GLOSSARY CYCLE - 'T-TeachText.OldLoc' begin at 10:17:51 AM 9/29/92 - 'AppleGlot 2.0'
End of 'T-TeachText.OldLoc' at 10:17:55 AM 9/29/92

INTERNAL CYCLE - 'T-Pr Update FDHD.OldLoc' begin at 10:17:56 AM 9/29/92 - 'AppleGlot 2.0'
End of 'T-Pr Update FDHD.OldLoc' at 10:17:56 AM 9/29/92

GLOSSARY CYCLE - 'T-Pr Update FDHD.OldLoc' begin at 10:17:57 AM 9/29/92 - 'AppleGlot 2.0'
End of 'T-Pr Update FDHD.OldLoc' at 10:17:57 AM 9/29/92

INTERNAL CYCLE - 'T-Battery.OldLoc' begin at 10:17:59 AM 9/29/92 - 'AppleGlot 2.0'
End of 'T-Battery.OldLoc' at 10:17:59 AM 9/29/92

GLOSSARY CYCLE - 'T-Battery.OldLoc' begin at 10:18:00 AM 9/29/92 - 'AppleGlot 2.0'
End of 'T-Battery.OldLoc' at 10:18:02 AM 9/29/92

VERIFY CYCLE - 'T-TeachText.OldLoc' begin at 10:18:03 AM 9/29/92 - 'AppleGlot 2.0'
End of 'T-TeachText.OldLoc' at 10:18:05 AM 9/29/92

VERIFY CYCLE - 'T-Pr Update FDHD.OldLoc' begin at 10:18:06 AM 9/29/92 - 'AppleGlot 2.0'
End of 'T-Pr Update FDHD.OldLoc' at 10:18:06 AM 9/29/92

VERIFY CYCLE - 'T-Battery.OldLoc' begin at 10:18:07 AM 9/29/92 - 'AppleGlot 2.0'
  'DITL' -15616            ??WARNING #37 Non-translatable field mismatch. TMPL expansion #5. 'DBYT' {Item type}.End of
'T-Battery.OldLoc' at 10:18:08 AM 9/29/92

UPDATE CYCLE - 'T-TeachText.OldLoc' begin at 10:18:29 AM 9/29/92 - 'AppleGlot 2.0'
End of 'T-TeachText.OldLoc' at 10:18:31 AM 9/29/92

UPDATE CYCLE - 'T-Pr Update FDHD.OldLoc' begin at 10:18:33 AM 9/29/92 - 'AppleGlot 2.0'
```

```
End of 'T-Pr Update FDHD.OldLoc' at 10:18:33 AM 9/29/92

UPDATE CYCLE - 'T-Battery.OldLoc' begin at 10:18:35 AM 9/29/92 - 'AppleGlot 2.0'
End of 'T-Battery.OldLoc' at 10:18:37 AM 9/29/92

VERIFY CYCLE - 'T-TeachText.OldLoc' begin at 10:18:38 AM 9/29/92 - 'AppleGlot 2.0'
End of 'T-TeachText.OldLoc' at 10:18:39 AM 9/29/92

VERIFY CYCLE - 'T-Pr Update FDHD.OldLoc' begin at 10:18:40 AM 9/29/92 - 'AppleGlot 2.0'
End of 'T-Pr Update FDHD.OldLoc' at 10:18:41 AM 9/29/92

VERIFY CYCLE - 'T-Battery.OldLoc' begin at 10:18:42 AM 9/29/92 - 'AppleGlot 2.0'
  'DITL' -15616           ??WARNING #37 Non-translatable field mismatch. TMPL expansion #5. 'DBYT' {Item type}.
End of 'T-Battery.OldLoc' at 10:18:43 AM 9/29/92
```

Appendix E AppleGlot’s Built-in Templates

The following resource templates (TMPLs) are built-into AppleGlot 2.0b15:

‘ALRT’	Alert template	
‘APPL’	Application list (Desktop)	
‘CNTL’	Control template	
‘DITL’	Dialog (or alert) item list	
‘DLOG’	Dialog definition	
‘errs’	Error string	
‘FBTN’	MiniFinder button	
‘finf’	Font information	
‘fld#’	Folder manager list	
‘FREF’	File reference	
‘icmt’	Installer comment	
‘inbb’	Installer boot block atom	
‘indm’	Installer default map	(obsolete: not used in Installer 3.1 and later)
‘infa’	Installer file atom	
‘inpk’	Installer package	
‘inra’	Installer resource atom	
‘insc’	Installer script	
‘MENU’	Menu	
‘PSAP’	Word-length labeled string	
‘STR ‘	Pascal string	
‘STR#’	Pascal string list	
‘TEXT’	Unlabeled string	
‘view’	View	
‘WIND’	Window template	