Foundation 1.0.2 Addendum, December 7, 1992

Additions & Changes to Foundation:

Page 10, ***Optimize File*** now allows the optional removal of Resource Names from the file. The dialog window shown in Figure 2-5 now contains a checkbox for selecting this option.

Page 10, ***Save*** documentation now should read:

> ***Save*** writes the active file directly to disk, making all changes permanent. If you are working with a file created with **New** that has not been saved previously, or one that can not be written to (because the file being locked or busy, disk locked or full, or any other error condition while trying to write), selecting **Save** will operate identically with **Save As...**. Foundation opens a file for read access only and copies it to an internal workfile- therefore a save actually rewrites the workfile to the file you originally opened/saved to.

Page 11, ***Paste*** has been extended to allow a pasted resource to replace another if it has a matching resource ID. A new dialog window will be presented to control how Paste should handle resources with matching IDs. There are four options for this new feature:

> ***Paste as a new item*** will add the single resource shown in the dialog to the current file and assign it a new ID. Each match will cause the dialog to reappear.
>
> ***Paste all items as new*** works like the first option, however any further ID matches that are discovered will be handled automatically and no additional dialogs will be presented to annoy you. This is the safest use of Paste and was the way Paste was handled in version 1.0.
>
> ***Paste over old item*** will add the single resource shown in the dialog in the current file, replacing the existing resource that has the same ID. Like ***Paste as a new item***, additional matches of other resources will cause the dialog to reappear. There are several things that you should be aware of when using this, or the next option:
>
> > 1) Dependent resources referenced by a replaced item may become orphans. This is especially true if the structure of an existing complex resource is different than the one you are replacing it with (example: *a 10 item menu with one having 8 items.*)
> >
> > 2) Replacing only part of a resource structure. (example: *keeping an existing window and its control list, but replacing a control does not guarantee that the control you replace is actually in that control list, and not another one.*)
> >
> > 3) Replacing only part of a resource structure may create orphans (example: *keeping an existing control list that had fewer controls than the one on the clipboard will result in the additional controls being added but never used.*)
> >
> > 4) Also, try to avoid multiply used resources... Various toolbox calls can be hurt or can hurt you, and we don't want to even think about the ramifications of this on Paste/replace.
>
> Generally, the best use of Paste/replace is with resources without dependents, like string lists, code resources and the like.
>
> ***Paste over all old items*** is the automated version of ***Paste over old item***, and the same warnings as to its use apply, however they should be underlined. To be honest, this option was added to balance the Paste options, and should be used only if you have a very good understanding of the resources that you are using.

Page 20 - 22 & Appendix A, ***ScriptEdit*** now includes ***ScriptBuilder***, which lets you create your own custom resource "scripts". ScriptEdit works as documented. However, this release of ScriptBuilder does not include the Link Tool, nor the operations Switch, Goto, Loop or the If... tests. These will be included in a subsequent release. One important note- if you are including an unformatted text block (one with no size delimiter), it needs to be at the end of the resource structure.

Manual Errata:
Page 6, Para 1, Line 10 - "ignominious" should read "nasty".

1.0.1 Addendum Errata:
Page 1, Para 1, Line 1- Errata - "devastating" should read "nasty".

The Foundation Developer's Kit now includes source code for a simple rPString (resource type $8006) Editor Module. The source is fully commented, and is provided in ORCA/C, ORCA/Pascal and ORCA/M Assembly versions. A Pascal String editor was selected because we could demonstrate how to work with the Foundation shell without having to get involved with a lot of code to actually edit the resource data. You may use these as a shell from which to build your own editors. (C & Pascal versions by Mark Collins & Marc Wolfgram, Assembly version converted by Andy Wells.)

Bug Fixes and Enhancements
There have been numerous bug fixes and other changes to Foundation in version 1.0.1 and 1.0.2. We wish to take this opportunity to thank all of our beta testers for their attention and interest. If you should encounter any problems or bugs, or have any suggestions for new features, please contact us.

Contacting Lunar Productions

There are several ways to contact us for help or information on Foundation...

Support areas on Online Services

America OnLine	(ADV)	Company Support Forum in the Apple II Developers Area
GEnie	(m530)	A2Pro Category 35- Lunar Productions Company Support Area

Online Mail addresses for Support

Applelink	D3672
Internet	D3672@applelink.apple.com
	mwolfgram@aol.com

Lunar Productions
1808 Michael Drive
Waukesha, WI 53186
(414) 549-9261 (evenings only, please!)

Thank you again for purchasing Foundation
Marc, Mark, Jim & Tammy