MacApp® 2.0b5

August 3, 1988

Welcome to MacApp 2.0b5. Much work has gone into MacApp since version 1.1.1 and some parts of MacApp 2.0 differ significantly from its predecessor. These release notes describe the major changes in MacApp.

The software in this package is considered beta quality. It has known bugs, and we're sure users of MacApp will find even more bugs. While we are not preventing you from shipping products based on this release of MacApp, please bear in mind that you do so at your own risk, with the understanding that this release of MacApp is not considered production quality. We encourage you to report bugs in the software or documentation, as well as suggesting improvements, by using the MacApp® Bug Report Form included in this package.

This release of MacApp is designed to work as is with MPW 2.0.2. It is also intended to work with the upcoming release of MPW 3.0 with little change. See the MacApp® 2.0b5 Feature Overview for more about using MPW 3.0. And please note that at the time of this writing MPW 3.0 is still changing so we cannot absolutely guarantee compatibility with the MPW 3.0 release that eventually is made available from APDA.

At this time the MacApp 2.0 Manual has not been completed. Even with these release notes and the MacApp® 2.x Manual (interim version) there are major portions of MacApp that are not documented. However, most of the differences between 1.x and 2.0 are documented in these release notes.

This package contains the following documents:

- An important **letter from Apple's Software Licensing Department** regarding distribution of MacApp and applications built with it.
- A copy of the **MacApp® Object Code Distribution Agreement.**
- **MacApp® Bug Report Form**, which contains the information necessary to report any software or documentation bugs you may encounter in MacApp.
- **MacApp® 2.0b5 Known Bugs and Problems**, which describes the known bugs and problems in this release of MacApp.
- **MacApp® 2.0b5 Delta List**, which describes the differences between this release and MacApp 2.0b2.
- **MacApp® 2.0b5 Feature Overview**, which describes the major new features of MacApp 2.0.
- **MacApp® 2.0b5 View Architecture Release Notes**, which documents the MacApp 2.0 view classes.
- **MacApp® 2.0b5 UTEView Release Notes**, which describes the contents of the UTEView unit and the TTEView class.
- **MacApp® 2.0b5 UDialog Release Notes**, which describes the contents of the UDialog unit and the various view classes contained therein.
- **MacApp® 2.0b5 UGridView Release Notes**, which describes the contents of the UGridView unit and its classes.
- **MacApp® 2.0b5 Printing Release Notes**, which describes how printing works in MacApp.

- **MacApp® 2.0 Memory Management**, which discusses memory management in MacApp.
- **MacApp® 2.0 Globals**, which describes global constants, variables, data types, and routines in MacApp.
- **MacApp® 2.0 Object and Method Reference**, which discusses object classes that also existed in MacApp 1.1.

To use MacApp we strongly recommend that you also have these products:

- ***MacApp® 2.x Manual (interim version)***, available from APDA. This is essential for all MacApp users. It contains an introduction to object–oriented programming and Object Pascal, an overview of MacApp's organization and architecture, a rudimentary cookbook with programming examples, and a description of MacApp debugging tools.
- ***Inside Macintosh Volumes I-V***, Apple Computer, Inc., Addison-Wesley Publishing Company. These books are essential for any Macintosh program, even with MacApp. They are available from APDA and many bookstores.
- ***MacApp® 2.0b5 Source Listings***, available from APDA. This package consists of cross-referenced source code listings of MacApp 2.0b5.

Additional software and literature that should prove useful to anyone using MacApp includes the following products:

- ***Object–Oriented Programming for the Macintosh***, Kurt Schmucker, Hayden Book Company. Available from APDA and in many bookstores. The description of MacApp is based on MacApp 1.x. However the description of object–oriented programming is still very applicable.
- ***Object–Oriented Programming: An evolutionary Approach***, Brad Cox, Addison-Wesley Publishing Company. Available from APDA and in many bookstores. A good introduction to object–oriented programming.
- ***Programming with Macintosh Programmer's Workshop***, Joel West, Bantam Books. Available from APDA and in many bookstores. A thorough description of MPW and an introduction to MacApp 1.x.
- ***MacApp Developer's Association Newsletter***, a regular publication filled with information of interest to MacApp users. A six issue subscription is available directly from the Association for $20 (or $30 outside the U.S.) MacApp Developer's Association, P.O. Box 23, Everett, WA, 98206-0023.
- ***MacApp Browser***, a handy desk accessory that makes it easy to read through Object Pascal source code (including MacApp and your own MacApp applications). Available from the MacApp Developer's Association and APDA.
- ***MacApp Object Libraries 1-8***, consisting of member contributed software based on MacApp 1.x. Available from the MacApp Developer's Association.

Using MacApp 2.0b5 With MPW 3.0

We have made modifications to use MacApp 2.0b5 with MPW 3.0. To use MPW 3.0 you must change the MacAppStartup file in your MacApp folder and set the MPW2 shell variable to false. MPW 3.0 is still changing so we cannot guarantee that this release of MacApp will be compatible with future release of MPW 3.0.

At the time of this writing the current release of MPW 3.0 (version a2) contains incomplete TextEdit interfaces. These interfaces will be completed in the next release of MPW 3.0. Until then you will have to modify the MacApp source file UTEView.p to include the missing TextEdit interfaces. Near the beginning of UTEView.p you will find the following code:

```
{$IFC qMPW2}        { TextEdit interface that isn't included in MPW 2.x. }
CONST
   doToggle                    = 32;

TYPE
   TEIntHook = (intEOLHook, intDrawHook, intWidthHook, intHitTestHook);

FUNCTION  TEContinuousStyle(VAR mode: INTEGER; VAR aStyle: TextStyle;
                                                hTE: TEHandle): BOOLEAN;
                                                INLINE $3F3C, $000A, $A83D;

PROCEDURE SetStylScrap(rangeStart, rangeEnd: LONGINT; newStyles: StScrpHandle;
                                                redraw: BOOLEAN; hTE:
TEHandle);
                                                INLINE $3F3C, $000B, $A83D;

PROCEDURE TECustomHook(which: TEIntHook; VAR addr: ProcPtr; hTE: TEHandle);
                                                INLINE $3F3C, $000C, $A83D;

FUNCTION  TENumStyles(rangeStart, rangeEnd: LONGINT; hTE: TEHandle): LONGINT;
                                                INLINE $3F3C, $000D, $A83D;
{$ENDC qMPW2}
```

For MPW 3.0a2 this should be changed as noted in boldface:

```
{$IFC qMPW2}        { TextEdit interface that isn't included in MPW 2.x. }
CONST
   doToggle                    = 32;

TYPE
   TEIntHook = (intEOLHook, intDrawHook, intWidthHook, intHitTestHook);
{$ENDC}

FUNCTION  TEContinuousStyle(VAR mode: INTEGER; VAR aStyle: TextStyle;
                                hTE: TEHandle): BOOLEAN;
                                INLINE $3F3C, $000A, $A83D;

PROCEDURE SetStylScrap(rangeStart, rangeEnd: LONGINT; newStyles: StScrpHandle;
                                redraw: BOOLEAN; hTE: TEHandle);
                                INLINE $3F3C, $000B, $A83D;

PROCEDURE TECustomHook(which: INTEGER; VAR addr: ProcPtr; hTE: TEHandle);
                                INLINE $3F3C, $000C, $A83D;

FUNCTION  TENumStyles(rangeStart, rangeEnd: LONGINT; hTE: TEHandle): LONGINT;
                                INLINE $3F3C, $000D, $A83D;
{Take out the $ENDC qMPW2 that was here}
```

MacApp 2.0b5 Known Bugs

This section describes known bugs and problems in the 2.0b5 release.

General Problems

1. MacApp is not yet as robust as we would like in low memory conditions.
2. MacApp has not been thoroughly tested with heap scrambling.
3. The samples tend not to be robust in low memory conditions.
4. The Pascal compile-time variables qTemplateViews, qProceduralViews and qWriteTemplates are not completely implemented and shouldn't be set to false.
5. Applications built with this release of MacApp will not run on a Macintosh XL (Lisa).

UDialog

1. Undo for TEditText and TNumberText is disabled because it had bugs that could cause the program to crash. It will be reinstated in a future release.
2. Invalid character entry in a TNumberText view produces a beep and not an alert. Since the text typed into a TNumberText view is not validated until the user tabs to another view or attempts to close the window, it is possible that the application will prevent the user from quitting if he has typed invalid text into a TNumberText view and the view is still selected. This behavior can be modified by overriding TNumberText.Validate or TDialogView.CantDeselect.

UGridView

1. Dim hiliting has been disabled because of bugs. It will be reinstated in a future release.
2. Resizing columns or rows causes the selected cells to be deselected.
3. FirstSelectedCell and LastCellSelected are not guarenteed to work unless the selection is a rectangle.

UInspector

1. Using the Inspector in low memory situations may crash the program.
2. The Inspector sometimes gets confused about which items are selected in the upper two panes. For example, if the class TList is selected and another TList object is created then the Inspector may get confused.

UList

1. The Delete operation in a TSortedList object does not use a binary search to find the object to be delete. Instead, it uses the same sequential search as implemented for TList. This will be corrected in a future release.

UMacApp.TCommand

1. There is a fundamental problem with command objects in that they don't have a notion of a target on which the command operates. This becomes a problem for undoable commands whose document is NIL. In that case MacApp assumes the command is an application command and will allow the user to undo it even if the context no longer makes since for the command to be undone. For example, suppose a command operates on a document-less view. Even if the view is freed the command will still be undoable until another undoable command is created.

UMacApp.TView

1. Shadow adornments do not print correctly.

UPrinting

1. It is possible to cause a crash by undoing the Page Setup command if the view associated with the command has been freed. This situation will most likely occur if a document has more than one window, the Page Setup command is used while one of the windows is active, then the window is closed.
2. The fEffectiveDeviceRes field of TStdPrintHandler is not always accurate. We suggest that you avoid relying on this field as we intend to take it out completely in a future release.

UTEView

1. The interface for TECustomHook is incorrect. The first parameter should be an INTEGER rather than the enumeration TEIntHook. The enumerated identifiers in TEIntHook should be made constants. This needs to be changed only if you use TECustomHook. It is not used by MacApp.
2. There is some flashing of characters as lines of text are added to a TEView (i.e. parts of characters on the line being added are displayed twice). This should be fixed in a future release.

UViewCoords

1. Pt2VRect does not work correctly if the horizontal or vertical coordinate of the first point is greater than the horizontal or vertical coordinate of the second point. It does work correctly if both coordinates of the first point are less than both coordinates of the second point, or both coordinates of the first point are greater than both coordinates of the second point. This routine is not used by MacApp, so it is not affected.

2. In the file UViewCoords.p we inadvertently left in a Pascal compiler include directive for UViewCoords.inc1.p, even though it doesn't exist. This generally will not cause problems as the UViewCoords.p unit is not compiled by MABuild. (The actual code for the routines defined in the unit is in assembly language.)

Changes Since MacApp 2.0b2

This document lists the differences between MacApp 2.0b5 and MacApp 2.0b2.

CommonObjLib.a

1. Fixed bug in which the non-optimized method table cache fragmented the heap.
2. The non-optimized method table cache now does a Macsbug break when there is a nil object or method. It used to do a division by zero to cause a break.

ObjLib.inc1.p

1. Fixed bug in which the non-optimized method table cache fragmented the heap.

UAssociation

1. Changed some segment mappings.

UBusyCursor

1. Changed name of watchCursor field in UBusyCursor.a to avoid conflict with MPW 3.0 interfaces.
2. Changed some segment mappings.

UDialog

1. TDialogView.DoSelectEditText and SelectEditText always select the characters when the selectChars parameter is true. Previously it didn't select the characters if the edittext view was the one being edited.
2. Fixed bugs in TStaticText and TEditText.SetTExt in which the text didn't get redrawn if it was set to an empty string.
3. Modified TDialogTEView.InstallEditText so that it sets its height to be at least as a high as a single line of text.
4. Change some segment mappings.
5. Fixed problems with nested clusters and radio buttons.
6. Added the methods TRadio.SetState and TCheckBox.SetState.
7. Added TDialogView.DoCommandKey to handle the command-'.' key combination.
8. The default control is highlighted if the users presses return. Similarly, the cancel control is highlighted if the user presses command-'.'.
9. Filled out the Fields methods.
10. Fixed bug in which compilation of TDialogTEView.Fields wasn't conditional on qDebug.
11. Fixed bugs in TStaticText and TEditText.Draw in which they weren't disposing of handles they created.
12. Added TStaticText and TEditText.DoSubstition.
13. Added TStaticText and TEditText.ImageText.
14. TPopup now works correctly with respect to color.
15. Added the global routine GetMenuColors, which returns the correct color values for any menu item or menu title.
16. Added TPopup methods CalcLabelRect, DrawLabel and DrawPopupBox.
17. Fixed other TPopup oddities.
18. TPopup.SetPopup now calls MAInsertMenu instead of the Toolbox routine InsertMenu to make sure the menu's color table gets installed.
19. Fixed bugs in TPopup.IPopup and TPopup.IRes in which the menu and resource id fields were not set if the resource id was kNoResource. Also, these routines make the popup invisible if the machine does not support popup menus.
20. Now check for presence of 128K ROMs before using TESelView and TESysJust.
21. Modified TDialogView.PoseModally to set fDismissed to false before handling events. This fixes a bug when calling PoseModally more than once on the same instantiation of a TDialogView.
22. Fixed bug in which a range check error could occur when doing string substitutions on static text strings.
23. Modified TNumberText.Validate so that it considers an empty string to be the number zero. Previously it considered an empty string to be invalid.
24. Removed the $B- compiler directive as it is not necessary.
25. Disabled undo in TEditText and TNumberText views because it had bugs that could cause the program to crash.

UGridView

1. Some code has been added to handle dim highlighting. However, it is not yet debugged so it has been disabled.
2. Added the method DoHilite to handle cell highlighting.
3. Fixed bug in InvalidateSelection in which it invalidated an extra row and column.
4. Removed InitUGridView. It used to register TGridView, TTextGridView and TTextListView,

but since these must be overridden to be useful, there is no point in registering them.

5. The SetPen method now sets the pen according to fTextStyle rather than taking a TextStyle parameter.
6. Fixed bug in TGridView.Free in which some handles weren't being freed.
7. SetRowHeight and SetColWidth now do nothing when attempting to set the height/width of a row/column to its existing height/width.
8. Added the DrawRangeOfCells method which is used to draw a set of cells from a top-left to bottom-right cell.
9. Added the methods FirstSelectCell and LastSelectedCell. Note that these are not guaranteed to work if the selection is non-rectangular.
10. Changed VPointToCell to return its result as a VAR parameter rather than as a function result.
11. Fixed bug in SetRowHeight in which the height was occasionally set incorrectly.
12. Fixed bug in shift-key selection in which the "anchor" cell was incorrect.
13. Fixed off-by-one bugs when redrawing.
14. Fixed various bugs related to selection highlighting.
15. Deleted the method HiliteCell.
16. Changed kDoHighlight to kHighlight.
17. Fixed bugs in DelRowLast and DelColLast when deleting multiple rows or columns.

UInspector

1. The Inspector views are created via templates if the compile-time variable qTemplateViews is true.
2. Now avoid inspecting an object if its address is invalid (i.e. odd).
3. Fixed bug in which inspecting the fClassesByID and fClassesByName fields of the Inspector document caused a crash.

UList

1. Changed segment mappings for TSortedList to be consistent with TList.
2. Removed TSortedList.Delete as it didn't work properly. (TList.Delete is used instead.)
3. For TSortedList.Compare and Search, added the constants kALessThanB, kAEqualB, kAGreaterThanB.

UMacApp

1. Made all global variables part of the MacApp interface.
2. Added FindStdView and RegisterStdView routines to support "standard" view types.
3. Avoid registering view types if the compile-time variable qTemplateViews is false.
4. Fixed bugs in GetFocus/SetFocus in which they did not save/restore gLongOffset.
5. Fixed bug in which the low memory global SysFontSize was accessed on 64K ROMs, where it wasn't maintained. This produced a bad font size in the gSystemStyle record.
6. Modified MacAppAlert to set gCursorRgn to an empty region to ensure that the cursor gets reset after the alert.
7. Now make use of compile-time variables qProceduralViews, qTemplateViews and qWriteTemplates.
8. Removed some old TFrame-related constants.
9. Renamed 'isVisible' parameters to 'redraw' for consistency.

UMacApp.TApplication

1. Fixed bug in OpenDeskAccessory in which an address error could occur.
2. The Clipboard window is now created via templates if the compile-time variable qTemplateViews is true.
3. Fixed bug in which gIdlePhase did not get set to gIdleBegin, thereby preventing

UloadAllSegments from being called.

4. TrackMouse now calls the command object's AutoScroll method to carry out autoscrolling rather than handling the scroll directly.
5. Modified GetEvent so that it doesn't call Idle on mouse-moved events if there are other events pending. (This is because MultiFinder effectly assigns a higher priority to mouse-moved events than other events.)
6. Modified GetEvent so that it calls Idle before calling WaitNextEvent if there are no events pending. This ensures that the menu bar and cursor are properly set.
7. No longer need to fake up activate events for desk accessories in HandleSystemEvent.
8. Updated the Fields method.
9. Changed the CommandKey method to DoCommandKey.
10. Moved non-target related menu setup code from DoSetupMenus to SetupTheMenus so that the code gets executed even if the target chain doesn't include the application object. This is mostly debugging code and enabling/disabling the Apple menu.
11. IApplication no longer looks for an 'mctb' resource whose id is that of the 'mbar'. Instead, you should use 'mctb' resources whose ids are the same as the menus you are installing and let the Menu Manager load the 'mctb' for you.
12. In GetRsrcWindow, the result of GetNewCWindow is cast to a WindowPtr.
13. Fixed bug in which the scrap didn't get exported when switching out to another application. The problem was that gClipWrittenToDeskScrap was set to false in SetClipView, when it shouldn't have been set to false if the clip view was gClipOrphanage.
14. In SetTarget, gCursorRgn is set to an empty region to ensure that the cursor gets reset.
15. Modified the failure handler in PollEvent to call HiliteMenu(0) to ensure that the highlighted menu title, if any, gets unhighlighted.

UMacApp.TCommand

1. Added the AutoScroll method, which can be overridden to change the way the command handles autoscrolling.
2. Added the document parameter to ICommand, making it explicit which document the command effects. Previously ICommand simply set fDocument to gDocument.

UMacApp.TControls

1. Fixed bug in which scroll bars were initially created as being active, whereas they are now initialized as inactive and made active when the Activate method is called. This bug manifested itself when opening more than one window at a time.
2. Fixed bug in which Control Manager controls were not dimmed when created from a resource which set the dim state to true.
3. Changed some segment mappings.
4. Added TControl.Resize to redraw the entire control when resized. Previously it only redrew the difference between the old and new size.
5. Fixed bug in TCtlMgr.DimState in which it ignored the redraw parameter.
6. Modified TCtlMgr.Hilite so that turning on hiliting sets the fCMgrControl's hilite to 10. This makes the control look as though the mouse is clicked in it.

UMacApp.TDeskScrapView

1. PICT data is now shown in preference to TEXT if both TEXT and PICT data are in the desk scrap.

UMacApp.TDocument

1. Removed the method GetPriviledges and the field fAccessMode. (The notion of access modes

is not being supported.)
2. All calls to OpenFile are now routed through the method OpenAFile.

UMacApp.TEvtHandler

1. CreateAView now calls FindStdView if the class name in the view's template is an empty string.
2. Added the DoCommandKey method so that any event handler can handle command-key events.
3. In CreateAView, added FailNIL after ShallowClone to catch failures in ShallowClone.
4. Added failure handling to DoCreateViews so that it frees the view hierarchy it's creating if a failure occurs along the way.

UMacApp.TScroller

1. Fixed bug in AdjustScrollBars in which the scroll bars didn't get redrawn properly if they changed size but didn't move.
2. AutoScroll no longer checks for the presence of scroll bars.
3. DoScroll now resets the value of gLongOffset if the scroller is the focused view. This fixes bugs in which gLongOffset might be inaccurate after scrolling.
4. DoScroll now makes sure the deltas are within range before proceeding.
5. Renamed 'isVisible' parameters to 'redraw' for consistency.

UMacApp.TView

1. Fixed bug in Resize in which it sometimes computed the frame size incorrectly.
2. Changed some segment mappings.
3. Added the redraw parameter to Show, and logic to redraw the view if the redraw parameter is true.
4. Modified DrawContents to improve performance.
5. AddSubView now sets the subview's fNextHandler field to the superview if fNextHandler is nil. Similarly, RemoveSubView now sets the subview's fNextHandler to nil if the subview's fNextHandler is the superview.
6. Removed some of the gIntenseDebugging messages.
7. ContainsPoint and DrawContents now check IsShown.
8. DrawContents passes the correct rectangle to Draw. (It may not have, depending on the previous version of MacApp you had.)
9. Changed Activate to use fHLDesired as the fromHL parameter to DoHighlightSelection. This fixed bugs in which the highlighting might have been incorrect when first displaying the window.

UMacApp.TWindow

1. Modified HandleMouseDown so that clicking an inactive window whose fDoFirstClick field is true causes the window to be selected *and activated* before handling the click.
2. SetResizeLimits now also sets the zoom size limit.
3. IRes now casts the result of NewCWindow into a WindowPtr.
4. Modified Resize to call SizeWindow without invalidating, then call INHERITED Resize with invalidation.

UMAUtil

1. Added the file UMAUtil.a which is a combination of the old UMacApp.a and UTEView.a files. Also moved ALoadMacAppSeg from UMemory.a to UMAUtil.a so that when linking an optimized 128K ROM program you no longer get the warning that UMemory.a wasn't needed.
2. Removed the routine LongerSide.
3. DefineConfiguration now uses System 6.x as the criteria for styled TextEdit.

4. FieldToString now calls the routine whose address is stored in gFieldToStrRtn. The default for gFieldToStrRtn is StdFieldToString. You may define your own FieldToString routine by stuffing its address in gFieldToStrRgn.
5. Add PBHOpenDeny and PBHOpenRFDeny support to OpenFile.
6. Got rid of the CPU constants which are no longer needed now that we use SysEnvirons.

UMemory

1. Modified LoadMacAppSeg so that when debugging it reports the number *and name* of the segment being loaded.
2. Added the routine LoadResidentSegments.
3. Changed some segment mappings.

UMenuSetup

1. Changed some segment mappings.
2. Now NullMenuProc sets the menu width to zero so we don't have to call MeedCalcMenuSize as much.
3. Added MAInsertMenu, a replacement for the Menu Manager routine InsertMenu, so that the menu's color table is added when the menu is inserted.

UObject

1. Removed the $B- directive as it is no longer necessary.
2. Changed some segment mappings.

UPrinting

1. Fixed bug in which turning off the page breaks didn't force an invalidation.
2. Changed some segment mappings.
3. Took out handling of update events in CheckButton. (Should be reinstated at a later date.)
4. In PoseJobDialog, now call gApplication.UpdateAllWindows to force the windows to get redrawn before continuing.
5. Fixed problems with page break computation in which the fixed size page meaning was reversed in the h and v directions.
6. Removed the B$- compiler directive.

UTEView

1. Added the redraw parameter to SetOneStyle.
2. Fixed bug in which auto scrolling didn't work. (The problem was that ScrollBy was called with the redraw parameter set to false.)
3. The prototype TEView is created only if the compile-time variable qTemplateViews is true.
4. Replaced calls to FieldToString, etc. with calls to WrLblField.
5. Replaced fInsetLoc and fInsetSize with fInset, a rectangle giving the inset in each direction. Changed the parameters to ITEView to reflect this. Now there are size and location parameters as for other views, and an insets parameter which indicates the top, left, bottom and right margins within the view.
6. Page breaks are now computed correctly for styled TextEdit views.
7. Added GetPrintExtent to return the view's extent minus the insets (margins).
8. Changed ContinousStyle so that you can specify the range of characters to be checked.
9. Changed CalcSelLoc to return a rectangle rather than a point.
10. The TextEdit interfaces missing in MPW 2.x are now included in UTEView.p rather than in a separate file.
11. TTEStyleCommand.RedoIt now simply calls DoIt.
12. ITEStyleCommand now sets up the mode for old TextEdit as well as styled TextEdit.

13. When pasting we now avoid calling SetStylScrap unless the TEView is styled.
14. Modified StuffText so that it doesn't set fSavedTEHandle to fText if they are equal.
15. Fixed bug in SetOneStyle in which it didn't set fHTE's lineHeight and fontAscent in the non-styled case.
16. Minor changes to segmentation.
17. Fixed bugs in which the selection didn't always get scrolled into view. This was particularly noticable when typing to the bottom of the view's window.
18. CalcRealHeight now correctly computes the view's height.
19. SetOneStyle now pays attention to the mode even in the non-styled case.
20. Updated the Fields methods.
21. Fixed bugs in which the use of TESelView and TEAutoView was not conditional on the presence of 128K ROMs.
22. Modified Resize so that it forces a redraw if fAutoWrap is true and the width and height of the view changes.
23. Fixed bug in BeInScroller in which it didn't call INHERITED BeInScroller. This caused the scroller to not reset its translation values.
24. Fixed bug in GivePasteData in which it didn't ensure the view was the styled type before trying to get 'styl' scrap data. This caused erroneous errors to be reported.

UTrace

1. Fixed bug in TRCException in which we ignored errors $7000 thru $7FFF instead of $7F00 thru $7FFF.

UViewCoords

1. All routines are now coded in assembly language.

MacApp Resource Files

1. Added 'res!' resources for the resident MacApp segments and resident debugging segments.
2. Added 'view' template for the Clipboard window.
3. Changed the TEView resource template to replace the old InsetLoc and InsetSize points with an Inset rectangle.
4. Got rid of the ListItem template as it was not used.

MABuild and Make Files

1. Modified to support MPW 3.0.
2. Make is now passed the -w option to suppress warnings. This prevents warnings if Make variables are defined in the application's make file as well as MacApp's.
3. Modified the segment mappings to prevent overload of the BBRes segment and to map TEInit to GInit.
4. Added the qProceduralViews, qTemplateViews, and qWriteTemplates variables to MacApp.make1. These are passed to the Pascal compiler and govern the compilation of view I-methods, and the IRes, WRes and WriteRes methods.
5. Added UTEView.p to UDialog's dependencies.
6. Fixed bug in which not all of the object files were dependent on the setting of the ROM128K flag.
7. Added the -bf option to the link options.

All Samples

1. Modified the 'SIZE' resources to be compatible with MPW 2.0 and MPW 3.0.

2. Modified the copyright notices in the "About" boxes to conform to Apple legal requirements.

Calc Sample

1. Calc now uses the GridView building block.
2. Calc now creates is views from templates.
3. Corrected path name to SaneLib.o in Calc.make.
4. Fixed various bugs.

DemoDialogs Sample

1. Added color to the popup menu example in the 'Views by Template' window.
2. Now use default class names in templates.
3. Set the appropriate targets for the various dialog examples. This fixed bugs in which keystrokes were not being handled properly by some of the dialogs.
4. Changed 'Nested Scrollers' to 'Side-by-Side' scrollers.
5. Added 'seg!' resource to reserve memory for the maximum code resource usage.
6. Popup menu examples are not created if the environment does not support them.
7. Increased the number of master pointers initially created to prevent heap fragmentation.

DemoText Sample

1. Mapped the ARes segment to Main so that the code in ARes is made resident.
2. Now use default class names in the templates.
3. Made the 'SIZE' and 'seg!' resources reasonable.
4. Shortened the prompt in the "Get Background Color" dialog, as the previous one didn't fit.
5. Fixed bug in which the font name of the selected font wasn't always checked.
6. Changed default font size to 12.
7. Fixed bug in which the low memory global TESysJust was used on 64K ROM machines, in which case it didn't exist.

DrawShapes Sample

1. Mapped the ARes segment to Main so that the code in ARes is made resident.
2. Corrected the 'seg!' resource.
3. Changed the 'mctb' resource id to match the corresponding 'menu' so that the Menu Manager automatically loads the 'mctb' resource.

Nothing Sample

1. Motified the 'SIZE' resource to be reasonable.
2. Modified TNothingView.Draw to do a PenNormal at the end.

PatView Sample

1. Corrected the "About" box and creator.
2. Added HandleFinderRequest override to prevent attempt to open files on Finder launch.
3. Now stagger the windows.

Puzzle Sample

1. Modified the MultiFinder partition size to be more reasonable.
2. Added a 'seg!' resource to identify the maximum code resource usage.
3. Added command-key equivalents to the File menu.
4. Fixed bug in "About" box.

TwoDocKinds Sample

1. Modified the MultiFinder partition size to be more reasonable.