

Volume

2

THE CLASSIFIED GROUP, INC.

SunType Classified Advertising Publishing Systems



Operation Manual

4.1 Release Notes

Disclaimer of Warranty

THIS SOFTWARE AND MANUAL ARE SOLD “AS IS” AND WITHOUT WARRANTIES AS TO PERFORMANCE OF MERCHANT ABILITY. THE SELLER’S SALESPERSONS MAY HAVE MADE STATEMENTS ABOUT THIS SOFTWARE. ANY SUCH STATEMENTS DO NOT CONSTITUTE WARRANTIES AND SHALL NOT BE RELIED UPON BY THE BUYER IN DECIDING WHETHER TO PURCHASE THIS PROGRAM.

THIS PROGRAM IS SOLD WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES WHATSOEVER. BECAUSE OF THE DIVERSITY OF CONDITIONS AND HARDWARE UNDER WHICH THIS PROGRAM MAY BE USED, NO WARRANTY OF FITNESS FOR A PARTICULAR PURPOSE IS OFFERED. THE USER IS ADVISED TO TEST THE PROGRAM THOROUGHLY BEFORE RELYING ON IT. THE USER MUST ASSUME THE ENTIRE RISK OF USING THE PROGRAM. ANY LIABILITY OF SELLER OR MANUFACTURER WILL BE LIMITED EXCLUSIVELY TO PRODUCT REPLACEMENT OR REFUND OF PURCHASE PRICE.

The Classified Group, Inc.
SunType Publishing Systems
PO Box 512108
Punta Gorda, FL 33951-2108
800-458-4605
Fax: 941-743-9096

[Http://www.classifiedgroup.com](http://www.classifiedgroup.com)

E-Mail: classgrp@aol.com

SunType Classified System and accompanying documentation:
Copyright ©1989, 1990, 1992, 1993, 1994, 1995, 1996, 1997,
1998, 1999, 2000 and 2001 by SunType Publishing Systems.
Copyright ©2001-2002 Classified Group, Inc.

All Rights Reserved: No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means electronic, mechanical, photocopying, recording or otherwise, without the prior written permission of The Classified Group Inc., DBA SunType Publishing Systems.

System users whose purchase of the classified system has been recorded Classified Group Inc., DBA SunType Publishing Systems are granted the right to make “working” copies of the program disks for use on a single computer at a single newspaper or magazine.

TABLE OF CONTENTS

CHAPTER 1

SUNTYPE CLASSIFIED

4.1 RELEASE

New Features for SunType's Version 4.1	1
Ad and Account Browsers	1
General Purpose Queues	1
Context-Sensitive Help	1
Logos and Classified Display Handling	2
Optional Live Ad Indexing	2
Selective Account Detail Views	2
Ad-Text Editing Facilities	2
Rate Sensitive Messages Queuing	3
Scripting Option	3
Password Protected	3
Override Flags	3
Permission Control	3
Automatic Bad Debt Write Off	3
Selective Production-Time Ad Scrambling	4
Ad Order Changing Tracking	4
Enhance Report Generator	4
Deadline Verification	5
Classification List By Name	5
Utility Editor	5
Windows Font Selection	5
Advanced Typesetting Format Setup	5
Run-Time Selection of Typesetting Drivers	5
Typographic Flexibility	5
Improved Hyphenation	6
Dual Language Hyphenation	6
Improved Spell Checking	6
Extended Memory	7
Features No Longer Supported	7
Installation and Startup of 4.1	7
Setup	8
SMSC ARec & ASub Master Account Files	8
Scripting	8
Printer specification	8
Account, Ad & Detail Deletion	9
Automatic Bad Debt Write Off	9
Deadline Lockouts	10
Live-Ad Indexing Option	11
Ad Browser Options	11
Hyphenation Options	12
Controlling Hyphenation	13
Selecting Hyphenation Language	14
Editing Exceptions	14
Placing Logos in Text Ads	14
Introduction	14
Setup	15
Use	15
Production Notes	16
New Typographic Options	16
Type Size Scaling	16
Condensing or Expanding Type	18

Automatic Leading	19
An Alternative to Automatic Leading	19
Negative Tracking	20
XPress® H&J Compatibility	20
Box Shading	20
Miscellaneous New Options	21
Phone Number Insertion Templates	21
Enhanced Mail and Voice Box Templates	21
Key-Lock Status Display	22
Suppressing Class Help Display	22
Expired Ad Locking	22
Per-Edition Scheduling	22
Credit Card Memory	22
Contract Tracking	22
Back-Charging Incrementals	23
Avoiding Zero TFN & Incremental Charge Items	23
Setting Receivability for Incrementals	24
Insertion Change Explanations	24
Word-Counting Options	24
Forcing Overstrike Mode	25
Utility Editor	25
End Program Query	25
Production Sort Options in 4.1	26
Locking Authorized Credit Cards	26
Customizing late fee and write off messages	26
Immediate Late Fees	27
Windows® Font Installation With SetFonts	27
Create a Program Item	27
Installation with SetFonts	27
Override Flags	28
Rate Table Enhancements	28
Long Variable Names	28
Multiple Typographic Fillfills	29
Queue Addition Directive	30
Beep Commands	31
Message Commands	31
Variable Names	32
DOS Extender	34
Memory managers	35

CHAPTER 2

CONTEXT-SENSITIVE HELP & DEFAULTS

Introduction	1
Field Defaults	1
Editing Help	2
Help File Structure	2
Title	2
Segmentation	2
Bold Lines	3
Paragraph Formatting	3
Help Contexts	4
Defaults	6
	8

CHAPTER 3

SELECTION-TIME AD SEQUENCE SCRAMLING

Introduction	1
Scramble File	1
How It Works	2
Notes on Reporting	2

INDEX

Classified 4.1 Release

6/7/1996

Chapter

1

1 *Features of SunType's Version 4.1*

➤ **Ad and Account Browsers**

From the account search screen, you may now browse lists of accounts or ads. Accounts are listed in the order determined by the search field the cursor is in when the browse accounts function key is pressed. Ads are in production order.

For both kinds of browsers, you may page or cursor through the list. You may also jump through the list by typing the first few characters of the name, phone number or class number and text.

➤ **General Purpose Queues**

A queue system allows you to build lists of ads, accounts or detail items to which you may wish to return.

Queues may be created or deleted by using the "Setup Classified Account, Ad & Detail Queues" option on the installation menu. Once queues have been created, ads, accounts or detail items may be added to a queue by pressing a function key while editing the account, ad or detail item. An operator may "work" a queue by pressing the Queues function key at the search screen. Choose an item from the queue and you'll immediately jump to it. After filing the changed item, you'll return to the queue list.

Items may be deleted from a queue by pressing the "Delete" key while the queue item is highlighted. Items appearing in the queue are still in the classified database. The queue is really just a list of items in the database to which you or another operator may wish to return. We anticipate the primary use to be for supervisor or credit review. When the next version of the classified system adds a "hold" flag for ads (*which will operate in addition to the credit hold mechanism*), ads will be placed automatically in a hold queue. Also, all the report generator modules can build queues of the ads, accounts or detail items selected. This means you could build resolicitation or questionable account status queues and work through them without printing a list and repeatedly searching.

➤ **Context Sensitive Help**

A help system is included with the classified 4.1 account, ad, payment and adjustment entry module. The help system pops up a scrollable help window whenever the operator presses the **F11** key. Pressing the **F12** key opens a help window and

jumps to help on the current entry field or mode. *(On a first-generation keyboard that may lack F11 and F12 keys, use Alt-F1 for general help or Alt-F2 for context-sensitive help.)*

The contents of the help system are customizable and offer options to set default or initial entries for many fields. Instructions for editing help contents are in a separate document, *(Context-Sensitive Help and Defaults, SunType's 4.1 Release Notes, Chapter 2)*

➤ **Logos and Classified Display Handling**

SunType's facilities make it easy to place graphic logos in text ads. The logos may be specified along with the ad text and are automatically printed in PostScript output or flowed with Quark XPress *(via ClassLink)* or World-Wide Web *(HTML)* ads. The logo facility is described in detail later in this document.

When a logo in a text ad just isn't enough, you'll want to use the classified system's display classified functions. These allow composition of classified display ads, whether in-column or multi-column, outside of classified. Classified tracks the classified display orders and can automatically include graphic files that make up the display ad in ad runs output to Post-Script, Quark or HTML. Classified also maintains a production queue of classified display orders and has a Windows application that helps use tools like Corel Draw or other illustration programs to compose the ads and update the classified display records. The Windows version of ClassLink also includes production queue management facilities. The classified display functions are separately documented in *(SunType's Classified Display Classifieds Feature is in Volume 4-ClassLink Xtensions, Chapter 3.)*

➤ **Optional Live Ad Indexing**

An option offers dramatically improved typesetting selection and proofing performance for sites that are using the zone option or keep large numbers of inactive ads on file. Some of these sites have experienced sluggish selection and proofing due to the fact that these modules had to wade through large numbers of ads that were inactive or in the wrong zone to find current ads in the correct zone.

This option is described in detail in a separate section later in this document. *(See page 12.)*

➤ **Selective Account Detail Views**

We've gotten lots of feedback that it can be very tedious looking through busy account detail summaries for particular types of items. Now, when account detail lists are displayed, you may choose to view: all items, ads only, invoices only *(open-item accounts only)*. In the invoice view, when you select an invoice and press enter, the invoice's detail list is displayed.

➤ **Ad Text Editing Facilities**

The ad-text editor from previous versions has been completely replaced. This version will handle ad texts of indefinite size and has searching, ASCII export and cut

and paste facilities — you can even cut, copy and paste between different ads. It's also generally slicker and should feel faster.

The new editor doesn't support Turbo Lightning spell checking. We recommend you use the new, built-in spell checker.

➤ **Rate Sensitive Messages, and Queuing**

The rate table programming language can now do a lot more than compute charges. It can be used to pop up messages to the operator, sound beeps and even place ads in queues.

The language has also been enhanced with long, more readable variable names and some useful variables.

➤ **Scripting Option for Running Paper or File Backup**

Sites with network or workstation reliability problems have told us that they'd like a running backup mechanism that still allows them to use the over-the-counter invoicing. This mechanism will write readable transcripts of new and changed accounts, ads, payments and adjustments to specified network or local files or devices. The mechanism does not interfere with over-the-counter invoicing.

➤ **Password Protected Ad, Detail and Account Deletion**

An operator with the correct password may now selectively delete ads, payments, adjustments and accounts. Accounts are rebalanced automatically if ads or detail are deleted. An audit log of deletions is maintained.

➤ **Override Flags**

This ad-entry module flags ads that have had their automatically computed charge, automatic scheduling, text count, sort key or inserts-to-date count overridden by an operator. It's easy to print reports on these ads.

➤ **Finer Permission Control**

It's possible to individually control permissions for different kinds of payments and adjustments. The old version only allowed you to set whether a user could add payments and/or adjustments in general. This version allows you to allow or disallow entry of cash, checks, credit cards, credits, other charges and memo items separately for each user.

➤ **Automatic Bad Debt Write Off**

The late fee and automatic write off module can automatically write off large bad debts as well as small balances. You can set the number of days, minimum and maximum balances that qualify for automatic write off.

➤ **Selective Production and Time Ad Scrambling**

Facilities added to the 4.1 version of the typesetting selection module allow edition-specific scrambling of ad sequence at the time of typesetting selection.

This facility was developed for customers with multiple products who wish to produce some editions with scrambled ad sequence and some without. Also, each edition may have some classifications that scramble together while others are produced in classification order. Ads may be tracked by classification even when scrambled.

Setup and use of this feature is described in a separate technical bulletin entitled (*Selection-Time Ad Sequence Scrambling, Chapter 3*)

➤ **Ad Order Change Tracking**

When a posted insertion order is changed in a way that changes the ad cost, the classified system automatically generates a new detail item for the credit or debit. Version 4.1 includes in that detail item information about what may have caused the cost change. The displayed change information includes the plus or minus changes in total insertions and text count. More information — including the operator making the change and the state of the ad when the change was made — is available by positioning the selection bar on the ad change detail item and pressing the enter key. All this information is also available through the report generator when printing detail-based reports.

➤ **Enhanced Report Generator**

Some very nice facilities have been added to this version's report generator. These include some variables, the ability to access ad text information from ad-based reports and the ability to access ad information from detail-based reports. The capabilities should make it possible to do much more complex invoices among other reports.

Reports are also no longer limited to 65,535 items.

And, several report generator modules may now be run from batch files with report definition files and output specified as command line arguments. This means that you can write a batch file to run multiple reports without operator intervention.

The ability to do string comparisons in ad-hoc selections and other expressions. This makes it much easier to print reports on a range of names, area codes, zip codes, etc.

Documentation for the 4.1 report generator facilities has been integrated in to the report generator appendix in SunType's 4.0 updated manual. Please ask for an updated copy of the appendix if you didn't receive one with this bulletin.

➤ **Deadline Verification on Ad Scheduling**

The classified system can “lock out” ad scheduling for editions that have already been typeset. This feature is described in detail below — including information on how to turn it off if you don’t want it.

➤ **Classification List By Name**

It’s possible to search for classifications by name as well as number. The search options are available when entering or editing the class field of the ad-entry form. (*To enable the alpha class list feature, go to the classification setup module on the Installation Menu and re-compile your classifications.*)

➤ **Utility Editor**

The utility editor is included for editing report definitions, classification lists and rate tables. The editor replaces MicroStar and is used automatically unless you specify otherwise. It has no limit on text size and features improved cut, copy and paste facilities.

➤ **Windows® Font Selection**

More and more users are typesetting or proofing via Windows. An included Windows program, SetFonts.Exe, allows setup of the classified font table using fonts available in Windows. (*See Chapter 3 or Volume 4 ClassLink Manual, Chapter 5*)

➤ **Advanced Typesetting Format Setup**

For the virtuosos out there, the typesetting format maintenance module offers the option of editing typesetting formats as a text file. This is meant to make it easier to do things like copy formats and macros from one typesetting table to another. Be cautious using this feature; check your changes after editing by calling up modified tables via the menu to confirm that the changes were parsed correctly.

➤ **Run Time Selection of Typesetting Drivers**

Earlier versions required that typesetting and proofing programs with appropriate typesetting drivers be copied into place before typesetting. This was a problem if multiple output methods were in use (*for example, proofing via PostScript® and doing final layout via XPress®*). The version 4.1 proofing and typesetting modules allow choice of PostScript®, ClassLink (*XPress*), Rich Text Format or Generic Typesetter output drivers at typesetting or proofing time. You can also specify default output drivers for each format table using the typesetting format setup program.

➤ **More Typographic Flexibility**

Version 4.1 will scale typefaces. So, where previous versions were limited to 20 fixed-size fonts, 4.1 allows for 20 different faces — each scaleable in tenths of a point. The vertical and horizontal scale of type may also be independently varied.

Complementing 4.1, the scaleable type is now an automatic leading command. When auto leading is turned on, the system automatically computes leading.

This means that type sizes may be changed without having to worry about changing leading. The automatic leading command may specify a percentage to be added to type sizes to allow for proportional white space between lines.

Another typographic refinement is the ability to do negative letter spacing to force tighter character spacing than provided for by font designers. There's also a typesetting command menu available within the ad-text editor portion of the ad-entry program that should make it easier for operators to change type sizes, faces and alignment.

➤ **Improved Hyphenation**

This version's ad-entry and typesetting modules incorporate a dramatically improved hyphenation method. There's also an option, described in a separate section below, to use the old hyphenation method if it's necessary to make sure that ad fits stay the same between versions.

To get an idea of how much better the hyphenation method is, we ran the old and new hyphenation routines against a list of 28,637 words compiled from a user's classified database. The newer hyphenation routine found nearly twice as many hyphen opportunities and made fewer errors. Comparing typesetting runs between old and new methods, the new routine inserted 30% more hyphenations in very narrow-column (*8.5 pica, with mostly 8-point type*) ad copy. **Note** that the advantages are nearly completely aesthetic: the new ad run was only one percent shorter than the old but it had many fewer loose lines.

➤ **Dual Language Hyphenation**

Version 4.1 also incorporates an excellent Spanish-language hyphenation. Language may be chosen by a typesetting code or macro, so that and Spanish ads may share the same database. The hyphenation language may even be automatically associated with rate tables or classifications so that it doesn't take any effort by ad-takers to choose the correct hyphenation method. **Other languages may be available; please let us know what you need.**

➤ **Improved Spell Checking**

The ad-entry program has a built-in spelling checker — eliminating the need for the memory-resident spell checkers employed by previous versions. The dictionaries include nearly 50% more words than version 4.0 dictionary and almost twice as many words as Borland's Turbo Lightning. Many of the words are commonly used proper nouns.

The spell checker also eliminates the need to maintain separate auxiliary dictionaries for each user (*and to periodically consolidate them*). Instead, users share a common supplementary dictionary which should be periodically cleaned up by a supervisor.

A maintenance module, described in a separate document, makes it possible to add and delete words from both supplementary and main dictionaries.

Removing SpellTSR from classified startup batch routines will activate the spell checker. Once you've satisfied yourself that the routines are working, you may delete the old dictionary file, STCDict, from the classified program directory.

➤ **Built-In Dos Extender Allows Program to Use Extended Memory for Faster Operation.**

The ad-entry program had gotten as big as it could within DOS 640k limits. Software technologies allow this version to combine the best features of overlays and DOS extenders. This version will run in less memory than the old, but can take advantage of extra memory to run faster than ever.

➤ **Features No Longer Supported in SunType's Version 4.0**

A few features of previous versions are no longer available in 4.1. Direct support for HP LaserJet® (*Plus and later models and their clones*) typesetting with fixed-size, downloaded fonts has been removed. That typesetting method was practical before the advent of inexpensive, high quality PostScript® cartridges but is really obsolete now. If you still wish to use a LaserJet, you'll need to upgrade it with a PostScript cartridge or use a Windows® program like Word or XPress® in conjunction with TrueType® or Adobe Type Manager®.

The support for AUXDICT.FIL as a system-wide auxiliary spelling dictionary has been removed. That mechanism was used to circumvent Turbo Lightning's limitations and is no longer necessary. The built-in spell checker has its own system-wide auxiliary dictionary and also allows you to add words to the main, compressed dictionary.

Please note SunType's Version 4.1 is currently supported there will be no bug fixes as of 1999 one must update to current version.

2 Installation and Startup of SunType's Version 4.1

Copy or unpack the files from the master program disks to your classified system program directory (*usually \suntype\classad or \smc\classad*). Copy the configuration file ClassAd.INI to your classified file directory (*usually \suntype\classad or \smc\classad*) or directories.

The typesetting selection files used by 4.0 and 4.1 have a different format. You must NOT update from 4.0 to 4.1 between a typesetting selection and typeset-

Currently there are four programs: XClass.Exe, which is an extended version of the ad-entry program; XCMenu.Exe, an expanded version of the menu program; Queue-Set.Exe, which may be used to create and delete queues; and UEdit.Exe, a utility editor that replaces the old ms.com program. Other programs are direct replacements for existing programs.

The 4.1 program will not run under 4.0 current classified system main menu (*CMenu.Exe*). Instead, they must be started as stand-alone programs or CMenu should be replaced with XCMenu.

The batch program file Runxclas.bat shows how to run the extended adtaking program.

3 Setup

Several setup parameters may be changed by editing the ASCII text file ClassAd.INI, which should be located in the classified file directory (*usually \smtype\classad or \smc\classad*). You may edit this file by running the “Set General System Characteristics” module on the installation menu and choosing option “8”. Also, you can edit the file with an ASCII editor like DOS 5+’s Edit, EdLin or the Windows notepad. If the file doesn’t exist, you may create it with an editor.

ClassAd.Ini should contain a listing of settings, one line per setting. Each setting should have the form “keyword=value”. Setting entries may be in any order.

3.1 *SMSC ARec & ASub Master Account Files*

If you have SMSC’s ARec or ASub programs, you should tell the classified system where to find the account with entries like:

```
ARec=f:\smc\aaa\arecf10a.a  
ASub=f:\smc\aaa\asubfaaa.a
```

If you have either, you should tell classified how to find the master zip code file with an entry like:

```
Zips=f:\smc\signfzpb
```

Note: SunType works with any open item accounting system.

3.2 *Scripting*

To setup backup scripting, add an entry like:

```
Script=c:\clscript.fil
```

This is where the name on the right is the full pathname of the backup script file you wish to use.

3.3 Printer Specification

The ad-entry program can use printer ports other than PRN for invoicing. To specify an alternative port, create an entry like:

```
PRN=LPT2
```

This will cause over-the-counter invoices to be printed with LPT2 rather than LPT1. To specify that invoices should be sent in the PostScript page-description language, use:

```
PRN=PostScript
```

The two kinds of entries may be combined:

```
PRN=PostScript;LPT2
```

If you need to override the printer type for a particular workstation, you may still use the "PRN" environment variable as specified in the version 4.0 manual. Environment settings will override ClassAd.Ini entries.

3.4 Account, Ad and Detail Deletion

The deletion capability must be setup by adding an entry like:

```
DeletePassword=PASSWORD
```

to the ClassAd.Ini file, substituting a real password for "PASSWORD". If this line is not placed in the file, the deletion facility is disabled.

When deletion is enabled, an ad or detail item may be deleted by placing the selection bar on the item (*in the detail summary screen*) and pressing Alt-D. Confirmation is requested. If there is no detail left in the account, Alt-D will request account deletion.

Summary information on deleted accounts, ads and detail items will be appended to the ASCII text file DeletLog.Fil (*which is automatically created*) along with the number of the deleting operator and date of deletion.

```
CAUTION: Routine use of the deletion facility is hazardous to the mental health of your bookkeeper and/or accountant! It's meant to be used only with careful consideration; that's why it's password protected.
```

3.5 Automatic Bad Debt Write Off

The Late Fee/Write Off module carries out automatic bad debt write off. Adding a line like enables it:

```
BadDebtWriteOff=90,50,1000
```

to ClassAd.Ini, where the first number is the number of days a balance must have aged to qualify for write off, the second number is the minimum dollar balance necessary to qualify and the third number the maximum balance. If the account is open item, all of these apply to the invoice and it is invoices, rather than accounts, that are written off.

When the late fee/write off module encounters a qualifying account or invoice, it will write off the qualifying portion of the aged balance — or the entire invoice — by inserting a credit. The account will be marked long term and no-credit. Also, when the aging module eliminates account or invoice detail, it will maintain a memo detail item describing the write off. The memo will remain so as long as the account is marked long-term. To eliminate the account, mark it transient and it will age out of existence.

When operating on an open-item account, the late fee/write off module will only write off invoices if there are no recent credits to the invoice. The bad-debt write off may be restricted to transient accounts by adding a “;T” to the directive line:

```
BadDebtWriteOff=90,50,1000,T
```

When setup this way, long-term accounts will *not* receive bad-debt write offs — only transient accounts will be considered.

3.6 *Deadline Lockouts*

When a new ad is entered, the classified system will no longer allow scheduling for editions that have already been selected for typesetting. This is ACCOMPLISHED by checking the date and edition of the last typesetting selection whenever new ad entry is started. This provides deadline protection that is flexible — allowing ad entry for an edition up to the last possible moment without having to set arbitrary cut-off times.

For this feature to work, your editions must be selected for typesetting sequentially. That is, editions must be selected in the order in which they appear in the scheduling table. You must not select an edition for typesetting before another edition that has prior scheduling. Also, if you select multiple editions on the same day, the edition selection must be ordered alphabetically. So, if editions “A” and “B” are scheduled for the same date, you should not select “B” before selecting “A”. The consequence of selecting out of sequence will be unnecessarily locking out scheduling of new ads for the editions that are out of sequence.

Adding the line below will turn off the deadline verification feature:

```
DeadlineLock=FALSE
```

to the ClassAd.Ini file.

Note that deadline verification is **not** applied when old ads are edited.

3.7 Live-Ad Indexing Option

If the live-ad indexing option is turned on, the classified system will maintain an index of live ads to speed typesetting selection, proofing and reports that need to access only live ads.

This option basically offers you a trade-off: in exchange for the classified system taking a little bit more time filing new ads — while it does the work to maintain the live index — you may get improved performance for typesetting selection, proofing and some reports.

The degree of improvement will depend on two factors: 1) whether or not your site is using multiple zones, and 2) the ratio of live/active to inactive ads on file. If your site is using the multi-zone option or if the ratio of inactive to active ads is high, you will nearly certainly benefit from this option. If you're not using the zone option and active ads are a substantial percentage of the ads on file, there will be little benefit — possibly even a net loss.

To turn on the live indexing option, you need to do two things:

Add the entry:

```
LiveIndex=TRUE
```

to the ClassAd.Ini file. Then, run the data integrity check (*maintenance menu choice "C"*) or rebuild the ad file (*maintenance menu choice "B"*), which will create a current live index file. From then on, the classified system's ad-entry, typesetting selection, emergency deselection, aging and purging modules will automatically maintain the index — adding new ads and removing expiring ads. The data integrity check and file rebuild modules will each do a complete rebuild of the live ad index when they run (*assuming, for the rebuild module, that the ad file is included in the rebuild list*).

```
AdBrowseIndex=TRUE
```

3.8 Ad Browser Options

You may not find ad browser as useful as it could be if your ads are not produced in alpha order within classifications. That's because the browser normally presents the ads in production order. If production order isn't alpha within classes, you won't be able to jump through the ad list by typing the class number and first few characters of the ad you're after. To fix this problem, add the line to the ClassAd.Ini file. This directive

causes the classified system to maintain an extra index of ads in alpha order within classes. Maintenance of this index will slightly increase system overhead whenever ads are filed and will increase the size of the ClassAd.Fil by several percent.

The browser index won't take effect immediately when the directive is added. You must first rebuild the indexes of the ClassAd.Fil by running the rebuild module on the Maintenance & Error Recovery module and selecting a rebuild of the ClassAd.Fil. This action will generate the index and enable the browsing method.

Another case where the default ad browsing technique may work poorly is if large numbers of ads start with the same or similar first lines. This is common in cases where the first line of an ad is standardized as some sort of keyword — like the make of a car, gender preference or kind of job. A browser that shows an endless string of "MAZDA" first lines won't help you find a particular ad. You may be able to cure this problem by adding the line:

```
AltFLine=TRUE
```

to the ClassAd.Ini file. This will cause the classified ad entry program to store the second — rather than first — line of ad text in the ad text field that's displayed when using the browser or reviewing an account summary. It will also be the text that will print in invoices and statements. This change will affect ads as they are entered or changed. It will not affect old, untouched ads. Also, it won't affect ad production order.

If you need to make this change on a classification-by-classification basis, then don't add AltFLine to ClassAd.Ini. Instead, mark each classification that you wish to use alternate first lines by adding the line:

```
FLAGS AltFLine
```

to the classification parameters entered via the "Create or Modify Classification Titles Module". For example, an auto class might appear:

```
:CLASS 100  
TITLE Autos For Sale  
SORT NUMERIC  
FLAGS AltFLine  
PROMPT  
year, make  
PB, PS, AC ...
```

4 *Hyphenation Options*

Version 4.1 incorporates a dramatically improved hyphenation routine. However, it also includes the old hyphenation routines and will use those routines automatically for

ads entered by previous versions. We think this will give users the best of both worlds: new ads will receive better hyphenation, but old ads will keep their original line-breaks and lengths so that billing is unaffected.

This behavior can be changed by setting the “41Hyphens” option in the ClassAd.Ini file.

If you wish to force the use of the hyphenation method for all ads —old and new — add the line:

```
41Hyphens=TRUE
Ad Browser Options
16 SunType Classified System
```

to the ClassAd.Ini file. This will cause old ads to be re-hyphenated with the newer method when they are typeset or edited.

If you wish to force the classified system to use the old hyphenation method for all ads — old and new — add the line:

```
41Hyphens=FALSE
```

to ClassAd.Ini. The only reason we can think of to do this is if you are mixing classified ad entry among workstations using 4.1 and 4.0 ad-entry programs and wish to assure uniform results.

4.1 *Controlling Hyphenation*

The hyphenation method also allows improved control. Each hyphenation opportunity has a penalty level reflecting the aesthetic quality of the hyphenation point. Penalty levels range from zero for a root word break to four for a break that’s phonetically correct but typographically unacceptable.

You can forbid penalty levels at or above a certain value with a code in the classified system’s typesetting format tables. Use the coding:

```
{J#}
```

Where “#” is a digit from 0 to 4 reflecting the penalty-forbid level you wish to set. For example, {J3} would forbid hyphens with a penalty level of 3 or 4. The default forbid level is 4.

Different hyphenation preferences can be set for different rate tables (*by putting the code in rate table start/end code strings*) or different classifications (*by putting the code in classification title typesetting strings*). So, it’s possible to aggressively hyphenate reader ads while cautiously hyphenating legal ads.

4.2 *Selecting Hyphenation Language*

The classified system version can hyphenate using English- or Spanish-language rules. Language may be selected with a typesetting code.

To select English hyphenation rules, use the code:

```
{U0}
```

For Spanish, use the code:

```
{U1}
```

To associate a language with a rate table, add one of the above codes to the rate-table on-string in the typesetting format setup. Be sure to use the off-string to switch back to your default language.

To associate a language with a classification or range of classifications, add one of the language codes to the appropriate typesetting string in the classification title setup. Remember to add the code to switch back to your default language in the next classification title.

A **note** on multiple-language use on PCs: The classified system is compatible with the DOS KeyB command. This facility allows operators to switchback and forth between using a standard keyboard for English and other languages. (*See the chapter on "Customizing for International Use" in the 4.0 manual.*)

4.3 *Editing Exceptions*

There's now an option on the typesetting format setup module's main menu to edit the hyphenation exception dictionary. When you finish an editing session using this option, the program will automatically sort the exception dictionary. At the same time, it will eliminate any exception specifications that match the hyphenation decisions the program would have made without the exception entry.

5 *Placing Logos in Text Ads*

5.1 *Introduction*

The classified system now makes it easy to include graphic elements like logos in classified text ads. Previously, most users handled these graphics by reserving space within ads, then pasting the logos into the ad run. The mechanism still requires that you prepare the graphics outside the classified system, but now references to graphic files may be inserted into ads without any operator coding.

The logo facility can include logos and other graphic elements that are saved as disk files in particular formats. If you print ads directly from classified to a PostScript

printer, graphic files must be in EPS (*encapsulated PostScript*) format. If output is via Quark XPress and ClassLink, any file format supported by XPress for pictures may be used. If you're outputting ads to the Web via the HTML drivers, GIF files should be used. If you're outputting to both XPress and PostScript, use EPS. If you're outputting to HTML in combination with either XPress or PostScript, prepare both a file for the printed output (*e.g., EPS*) and a matching (*same filename except for the three-character type extension*) GIF file.

5.2 Setup

You should pick and create a directory to serve as a logo library. You may wish to make it the "logos" subdirectory of the classad (data) directory. If the logo files in use by classified are all in one place, it will make it much easier for operators to find them. Tell classified the location of the logo directory by adding an entry like:

```
LogoDir=\suntype\classad\logos or LogoDir=\smc\classad\logos
```

to the ClassAd.Ini file. If the entry doesn't exist, classified will look first in the logos subdirectory of the classified data directory.

You may also wish to constrain users to certain classes of graphic files. You may do so by listing all acceptable graphic file format extensions in a classAd.Ini entry like:

```
ValidGraphics=EPS,TIF
```

This line advises classified that EPS and TIF (Tag Image Format) files may be specified for logos. EPS, since it's listed first will be suggested as a default. If you don't make a "ValidGraphics" entry in ClassAd.Ini, only EPS files will be accepted.

5.3 Use

To add a logo to a classified ad, edit the ad's text and pick the line at which you wish to add the logo. Then, press **Alt-L** or choose "Graphic/Logo" from the Insert Menu **F8**. This will pop up the editor's "Logo Wizard" which helps format the logo without coding. The Logo Wizard asks for three pieces of information about the logo: its desired depth, an identifying tag and the filename of a graphic file.

The **Depth** is the desired vertical measurement of the printed logo. The unit of measurement defaults to points, but you may also specify inches or picas by adding "in" or "pi" to the measurement you enter. When the ad is printed or measured, this much vertical space will be allotted for the graphic. If the graphic is relatively tall or narrow, it will fill the vertical space and be centered horizontally in the column. If the graphic is relatively wide, it will be centered in this vertical space and fill the column width.

The **Tag** should be used to identify the logo or graphic. If, at type setting time, a graphic file isn't available the tag will be used to help composition workers fill the absence.

The **Print File** field should receive the name of the graphic file associated with the desired logo. If you enter a filename with one or more wildcard characters — like “*.eps” — a list of all matching files will be displayed for selection. If no print filename is entered, the various output drivers will just send the tag in an empty box. In that way, you can use the logo facility to reserve vertical space for a graphic to be added at production time.

When you finish with the Logo Wizard dialog, it will insert logo code into the classified ad. That code will always be on a line by itself and have the format:

```
{!depth,print_file_name,tag}
```

In general you won't want to edit these codes since they're complicated. Instead, just delete the whole code and use the Logo Wizard to create a new entry.

After the logo code is entered, the classified system will count the logo's depth in all measurement and pricing calculations.

5.4 *Production Notes*

If you're using the logo facility in conjunction with ClassLink to flow ads into XPress, make sure you read the “Gather Files” section in the *Classified Display Features* document. It covers issues that you should be aware of about making the graphic files available to XPress.

Another issue for XPress users: Logos and other graphics are passed to XPress as *anchor boxes*. This can pose a problem, because that's also how ClassLink handles boxed ads. The problem is that XPress doesn't allow anchor boxes within anchor boxes. So, if you try to put a logo inside a boxed text ad, it won't pass to XPress. (*It should print correctly via the classified PostScript drivers.*)

Finally, if you're using the HTML drivers to produce world-wide webpages, but using other drivers to produce print pages, you should create both EPS and GIF (*Graphic Interchange Format*) versions of logo files. The files should have identical file and pathnames except for the three character file-type extensions. Supply the name of the EPS file for the print file name. The HTML driver automatically looks for the matching GIF file.

6 *Typographic Options*

6.1 *Type Size Scaling*

Previous versions of the classified system supported only a limited number of typeface/size combinations. Version 4.1 offers a command to scale typefaces.

In order to use this feature for the first time, you must use the typesetting format

setup module (*Installation Operations/ Create or Modify Typesetting Format Specifications*) to edit the font table. Font scaling information will be automatically installed by choosing the edit option. Type sizes specified in the font table set the **default** size for a typeface.

Once font scaling information has been created, you may use the type size command in typesetting formats, macros and within the ad-text editor. Type sizes and faces may also be chosen from menus in the ad-text editor (*press **F10** for the main menu*).

The type size command is like other curly brace formatting commands. To set type size, enter a code string like:

```
{N12.5}
```

This command changes the type size for the current typeface to 12.5 point. The command:

```
{N8}
```

would change type size to eight-point.

Note that whenever you change type faces with the {F#} command, you are also setting the type size to the default size for the type face that's specified in the font table. So, a command like:

```
{N12F3} WRONG!
```

is a bad idea because the type size command "N12" will be overridden by the font command "**F3**". Instead, the command:

```
{F3N12} RIGHT
```

Sets the font to number three and scales it to twelve-point. The way that a font choice overrides a size specification is a behavior meant to maintain compatibility with older versions of classified. If it didn't work this way, then many older typesetting format tables wouldn't work right. If you're starting with a clean slate, you may use the command:

```
{“1}
```

To change the bias so that size specifications are not overridden by font choices. Typically, you'd place this code string in the initialization string for the typesetting format table. You can reverse the command:

```
{“0}
```

Will restore the default font/size choice behavior.

6.2 *Condensing or Expanding Type*

If you typeset your ads via PostScript or ClassLink to XPress, you may now squeeze or expand typefaces by scaling the faces height independently from its width. To do so, first set the face's width with the size command. Then, set the vertical scale with the “{E##.#}” command. For example, to create a font seven points high and with a set width of 14 points, use the command:

```
{N14E7}
```

Note that the font size command “N” takes precedence over the vertical scale command “E”. So, if you use a command like:

```
{E7N14} WRONG!
```

the “N14” will override the “E7” and the result will be 14 points wide and tall. You may prefer to specify horizontal rather than vertical spacing. The “{#x.x}” command allows you set a horizontal type size. For example:

```
{N7#14}
```

Will produce type seven-points high and with a horizontal set size equivalent to 14-point type. It's the same result as “{N14E7}”.

Please note, for either command:

Electronically squeezed or expanded type can produce dramatic effects for class headers but should nearly always be avoided in ad body text where it may reduce legibility. If you need a tighter copy fit, use a face that's been designed for that purpose rather than squeezing an existing face.

NOTE: The classified system's RTF driver doesn't support independent vertical scaling since this feature is not included in the RTF standard.

6.3 Automatic Leading

A automatic leading command complements the type scaling. When automatic leading is turned on, leading will be automatically varied with type size changes. This allows easy changing of type size within ads without worrying about line spacing.

To turn on automatic leading, use the command:

```
{A1}
```

Any fixed leading command or the command:

```
{A0}
```

turns off automatic leading.

You may set the auto leading feature to automatically add a certain percentage of extra space by putting the percentage in the command argument. For example, to add 20% of extra automatic leading, use the command:

```
{A20}
```

How It Works

The automatic leading facility works on a line-by-line (*vs .paragraph-by-paragraph*) basis. Leading for each line is set to 25% of the largest size on the previous line (to account for descenders) plus 75% of the largest type size on the current line. Extra automatic leading is computed from this value. Remember that classified system leading is computed from baseline-to-baseline and takes place **before** printing a line.

If you use ClassLink to XPress, results may vary since XPress leading facility works on a paragraph-by-paragraph rather than line by-line. Avoiding type size changes inside paragraphs will minimize this effect. A fix for this incompatibility may soon be available for XPress versions 3.3 and later.

6.4 An Alternative to Automatic Leading

If you don't want to use automated leading, but still want to make it easier for operators to change type sizes, you may add the line:

```
AddLead=TRUE
```

to the ClassAd.Ini file.

This option changes the way the ad-text editor's type size command C-N behaves. Normally, choosing a type size from the size menu will insert a formatting command like "{N12}" if you choose 12-point type. When the Add Lead option is turned on, though, "{N12S24}" (*12-point type, 24 half-points of leading*) is inserted instead.

6.5 *Negative Tracking*

Version 4.1 allows for negative tracking (*letter spacing less than the design width of a typeface*) to reduce white space between characters.

Negative tracking may be set in 1/100 em spaces with a command like:

```
{T-2}
```

This command would reduce the letter spacing for each character by two one-hundredths of the current em-space width (*which is always equal to the current set width*).

Tracking commands persist until they are turned off (*with a "{T0}"*) or changed.

6.6 *XPress® H&J Compatibility*

The hyphenation and justification (*H&J*) techniques used by Quark XPress® differ slightly from those used normally by the classified system. If you intend to use XPress to paginate classified ads, you should add the line:

```
XPRESSHAJ=TRUE
```

to the ClassAd.Ini file to insure compatible line-ending decisions. When this command is in place, the ad-entry, typesetting and proofing modules will all adjust their H&J methods to improve Quark compatibility.

The differences from the classified system's default H&J methods when this switch is in use are slight; only a very small percentage of ads are likely to be affected. Affected ads will increase in lineage.

The default for this option is "OFF" or "FALSE" to retain compatibility with earlier versions of the classified system.

6.7 *Box Shading*

A typesetting format command allows specification of background tints (fills) for boxes. Only the ClassLink output driver currently honors the box background command. To specify a background tint, use a command like:

```
{&15}
```

to specify a percentage fill. The example command sets a 15% shade of black.

7 Miscellaneous Options

7.1 Phone Number Insertion Templates

The way the ad-text editor inserts phone numbers into ad texts when Alt-F is pressed may now be customized. You may set the format of local area code and non-local area code phone numbers by adding lines like:

```
LocalPhone=###-####  
LongPhone=(###) ###-####
```

The editor will insert a phone number digit in place of each “#” character. If there are only seven #s in the template, then the area code will be omitted— as in the local phone number example. If there are ten, the area code will be included.

Note that this provides away to create non-breaking phone numbers (*ones that won't be broken onto multiple lines*) if you need them. Just use then-dash (*ASCII 196*) in place of a hyphen.

7.2 Enhanced Mail and Voice Box Templates

Mail and voice box insertion templates allow you to specify how mail and voice box information should be inserted into classified ad texts. These have been enhanced in 4.1 to include the ability to insert ad expiration dates and to offset inserted dates by a specified number of days. They've also been changed slightly to allow use of the insert space command in the templates.

In both kinds of templates, it's now possible to place an underscore “_” character where you wish to insert the ad expiration date. So, for example, you may specify a mailbox template like:

```
^Box #=, expires _.
```

And, this would translate to “^Box #13112, expires 12/01.” in an ad that expired on December first. The date offset feature allows the specification of a template like:

```
^Box #=, expires _+15.
```

which would expand to “^Box #13112, expires 12/15.” in an ad that expired on December first. Date offsets may also be negative by using the minus “-” sign rather than plus sign.

7.3 *Key-Lock Status Display*

```
ShowCap=TRUE
```

This will enable display of caps lock, num lock and insert status displays.

7.4 *Suppressing Class Help Display*

The classification help display can take up as much as ten lines of the adtext editor screen. Now that this information is available as pop-up help (*press **F11** in either the ad or ad-text screens*), you may wish to suppress this display. To do so, add the line:

```
ClassPrompt=FALSE
```

to the ClassAd.Ini file.

7.5 *Expired Ad Locking*

The classified system will forbid update of expired ads if the line:

```
LockExpired=TRUE
```

is added to ClassAd.Ini.

7.6 *Per-Edition Scheduling*

```
AskPerEd=TRUE
```

This enables edition-style scheduling where the inserts ordered field will be interpreted as meaning inserts per edition rather than total number of inserts.

7.7 *Credit Card Memory*

Credit-card memory will be disabled if the line:

```
UseCardMemory=FALSE
```

is added to ClassAd.Ini. The default behavior is for the classified system to remember the most recently used credit card for each account and fill in that information whenever a new credit card payment is created.

7.8 *Contract Tracking*

Contract tracking will be enabled if the line:

```
TrackContracts=TRUE
```

This is added to ClassAd.Ini. This feature is meant to help track a series of ad duplications as a single contract. When this feature is enabled, every ad will automatically receive a notation like “CN#0032” in its note field when the ad is saved. The number in the note is the account item number and is different for every new ad. However, if an ad is duplicated and the contract number of the original is left in place in the note field, it will be left unchanged. If an operator wishes to break the contract chain — making a duplicate ad into a new contract— he or she should clear the contract number from the note field before filing the duplicate.

If there is anything except a contract number in the note field when an ad is filed, the contents of the note field will be pushed right to make room for the contract number. This will result in truncation of the contents if there isn’t enough room for both the old note and the contract number.

The report generator may be used to track and test contracts by using “AcctCode,Note,ItemNo,SubNo” for a sort specification and using the “.SETSUB” and “.IFSTREQ” directives to test for contract number changes.

7.9 *Back-Charging Incrementals*

When incrementally charged ad orders are changed, the classified system does not ordinarily generate an automatic adjustment to inserts that have already been billed (*unless the ad has been charged in advance, in which case the system adjusts for inserts billed but not yet made*). The rationale for this is that most customers using incremental charging are doing so because they want to “freeze” the past and avoid changing charges already billed.

If you want to change this behavior and have the classified system automatically adjust old incremental charges (*back-charging or back-crediting*), add the line:

```
IncChargeBack=TRUE
```

to the ClassAd.Ini file. The adjustments will be made in an automatically generated, single, compensating entry.

7.10 *Avoiding Zero TFN's and Incremental Charge Items*

The classified system’s “Log Automatic Receivables” module normally will generate incremental or TFN charge detail items when an ad qualifies for automatic charging even if the increment or TFN charge is zero. This is a feature meant to help track insertions when a precise audit trail of insertions is required for even free ads. However, this behavior is undesirable in many situations because it generates large volumes of unnecessary detail. To prevent creation of automatic items with zero charge, add the line:

```
AddZeroItems=FALSE
```

to the ClassAd.Ini file.

7.11 *Setting Receivability for Incrementals*

You may now set different receivability criteria for incrementally charged orders than for lump-sum charged ad orders. To do so, add a line:

```
IncReceive=Immediate | First | After
```

Specifying *one* of the choices: “Immediate” for immediate receivability, “First” for first-insert receivability or “After” for charging each increment after it’s completed.

Note how receivability works with incrementals. “Immediate” charges each increment in advance of its actual insertion; “First” charges the increment after the first insert of the increment has taken place; and, “After” charges each increment on its completion.

7.12 *Insertion Change Explanations*

If you add the line:

```
AskChangeNote=TRUE
```

to the ClassAd.Ini file, the ad-entry program will ask the operator for a brief, explanatory note whenever a posted ad order is changed in a way that changes the ad cost. The note will be displayed with the insertion change detail item and is available to the report generator when printing detail-based reports.

Otherwise, the classified system will use the text “Ad Order Change” as the explanatory note for insertion changes.

7.13 *Word-Counting Options*

Version 4.1 automatically remedies a couple of problems that earlier versions had counting words:

- **Words compounded by the slash mark — for example “his/her”— are now counted as two words unless the program thinks they may be part of a fraction.**
- **Insert-space and dot-leader codes — “^” and “|” — are counted as word divisions.**

These counting rules are only applied to ads that are entered under this or later versions of the classified system. So, if you’re upgrading to 4.1, ads created under version 4.0 or earlier will keep their old counts.

If you don't like these rules, all is not lost. To turn off the slash rule, add the following to ClassAd.Ini:

```
SmartSlash=FALSE
```

To turn off the rule for insert-space and dot-leader codes, add:

```
ISisSpace=FALSE
```

7.14 *Forcing Overstrike Mode*

By default, the classified system's insert mode is "sticky." When you move from field to field in an on-screen form, the program remembers whether its last editing mode was inserting or over-striking characters. This seems natural to some but crazy to others, who argue that the field editor should always start in overstrike mode whenever a new field is entered.

To force the classified system to revert to overstrike mode whenever it enters a new field in an on-screen form, add the line:

```
ForceOverstrike=TRUE
```

to the ClassAd.Ini file.

7.15 *Utility Editor*

The text editor used by the report generator, classification setup and rate table setup modules may now be specified with a ClassAd.Ini entry. Use a line like:

```
editor=edit.com
```

to specify an alternative editor. The text editor should be able to open and save ASCII text files and must accept the name of a file to be opened as a command-line argument. If no alternative editor is specified, UEdit.Exe will be used. (*The version 4.0 method of specifying an editor via an environment variable also still works.*)

7.16 *End Program Query*

A frequent request from users was to make it harder to accidentally exit the classified entry module by pressing the **Esc** key one too many times. The 4.1 version asks if you really wish to exit the program. This will no doubt irritate some users to no end. So, you may eliminate the query by adding the line:

```
AskExit=FALSE
```

to the ClassAd.Ini file.

7.17 *Production Sort Options in 4.1*

- **Two production sort options are available in version 4.1:** · Within classes, sort all charged ads above all uncharged ads and randomize ads with the same charge status.
- **Sort ads within classes alphabetically with six characters significant, then by ad size with largest ads first.**

Either sort order may be set as default using the Set General System Characteristics module or chosen for particular classes via the classification setup module. Using the latter, the charged ads sort may be chosen with the keyword “Paid”. The alpha/size sort may be chosen with the keyword “AlphaSize”. For example:

```
CLASS 10  
SORT Paid  
TITLE Real Estate For Sale
```

:
in the ClassTbl.Fil sets up classification 10 with the charged ads sort.

7.18 *Locking Authorized Credit Cards*

The version 4.1 ad-entry module locks credit-card-charge detail items with filled-in authorization code fields so that they may not be edited. They may still be viewed, just not changed. This fills a security gap in previous versions that allowed changes to credit card items after authorization.

If you don't like this new feature, you may turn it off by adding the line:

```
AuthCardLock=FALSE
```

to the ClassAd.INI file.

7.19 *Customizing Late Fee and Write Off Messages*

The text printed for late fee and automatic write off detail items — both on screen and in statements — may now be customized by adding lines to ClassAd.Ini. An entry like:

```
LateText=Gentle Punishment  
SmallWriteText=Who counts the pennies?  
BadWriteText=Never Again!
```

Replaces the late fee text with “Gentle Punishment”, the small-balance write-off text with “Who counts the pennies?” and the automatic bad-debt write-off message with “Never Again!”

7.20 *Immediate Late Fees*

Prior versions of classified would not assess late fees during the first aging period. This posed a problem for users who wanted to use the late fee mechanism to charge billing fees. To enable late fees before an aging period has passed, add the line:

```
LateAtZero=TRUE
```

to the ClassAd.Ini file.

8 *Windows® Font Installation with SetFonts*

8.1 *Installation with SetFonts*

SetFonts.Exe is the Windows program that allows editing of the classified font table from inside Windows using the Windows font list. You may use it to install fonts for classified if you are using a Windows program like Quark XPress® for classified output.

To install SetFonts follow the *Microsoft Windows User's Guide* instructions for creating a program item or just run the program from the Windows File Manager.

8.2 *To Create a Program Item by Using File Manager*

- Locate the Program Manager group where you want to create a new program item. The destination group can be either a window or an icon.
- Open File Manager and select the program or document file that you want to add as a program item. (*You can also select multiple items and add them to a group at the same time.*)
- If necessary, reposition the File Manager window so that you can see both File Manager and the Program Manager group icon or window where you want to place the new program item.
- Drag the file icon from File Manager to Program Manager, and drop it in the group window or onto the group icon. As you drag the file icon, the mouse pointer turns into a replica of a file icon. If you select multiple items, you can drag them all at the same time.

Note: If no other application windows are open except for File Manager and Program Manager, you can press CTRL+ESC to open Task List, and then choose the Tile button to arrange the File Manager and Program Manager windows.

9 *Override Flags*

Whenever an ad's automatically computed charge, text count, automatic scheduling, sort key or inserts-to-date fields are overridden by an operator, the ad-entry program sets a hidden flag to record the override. These flags are easy to test for in the classified system's report generator.

Some of the flags are in the "Special" variable. The special number for each is:

Charge Override 23
Count Override 22
Ins Made Override 21

So, to print a report of all ads with charge overrides, use the selection criteria:

23 IN Specials

Other override flags are in a "Flags" variable. Their numbers are:

Skip Scheduling **1** (*ad is skip-scheduled*)
Sort-Key Override **3** (*automatic sort key overridden by seller*)

10 *Rate Table Enhancements*

10.1 *Long Variable Names*

In previous versions, rate tables could be hard to read because the rate table compiler only recognized single-character names for variables. So, you had to remember that "C" meant charge, "K" meant credit status and so on. The 4.1 rate table compile will now recognize long variable names as well as the old, single-character names. That means that you can write code like:

Charge := (TextCount - 15) * 0.25 * InsOrd

in place of:

C := (W - 15) * 0.25 * I

The long variable names are listed at the end of this section.

10.2 *Newer Variables*

➤ **Module**

The “Module” variable is preset to tell you which program module is running at the moment. The possible values are:

1	Ad-Entry
2	Log Auto Receivables
3	Late-Fee/Write-Off

➤ **IsNew**

The “IsNew” variable is preset to have one of two values:

1	This is a new ad
0	This is an old ad

A new ad is one being freshly entered into the ad-entry module. An old is an already entered ad that is being edited.

➤ **ChargePerIns**

The “ChargePerIns” is preset to contain any previously computed charge per insertion value for an ad. It’s generally zero or meaningless for a new ad.

➤ **Field**

The “Field” variable is a write-only variable that may be used to force the ad-entry program to return to a particular field on the ad form. This variable would generally only be used to provide data validation beyond what’s ordinarily done by the classified system.

For example, let’s say that you do not wish to allow operators to use a classification 100 when scheduling reply box ads.

The code:

```
IF (Class = 100) AND (BoxType > 0) THEN  
Beep 1  
Message “Don’t use class 100 with reader reply ads!”  
Field := 6  
ENDIF
```

If the specified criteria are met, an error noise will sound, the message will be displayed and the entry cursor will jump to the class field.

Use this facility carefully. Once you've forced the entry cursor back to a field, the operator needs to know how to correct his or her error. If the error can't be corrected, the operator will be stuck! The cursor will repeatedly return to the same field until the operator aborts ad entry. So, make sure you use a message that helps the operator correct the error.

The field numbers for use with the "Field" variable are:

1	Ad taker (<i>operator number</i>)
2	Start date
3	Inserts ordered
4	Editions
5	Class
6	Rate table
7	Specials
8	Ad text
9	Note string
10	Voice box
11	Sort specification
12	Text count
13	Charge

10.3 *Queue Addition Directive*

The "Queue" command may be used to put the current ad in one of the classified system's general-purpose queues. The form of the command is:

```
Queue QueueName "Note"
```

QueueName should be the up-to-eight-character name of the target queue. The note should be text to be displayed in the queue along with basic ad information.

For example, let's say you wish to have all new ads that price out to over \$100 added to the "SupRev" supervisor review queue. The code:

```
IF IsNew AND (Charge > 100) THEN  
Queue SUPREV "New ad over $100"  
ENDIF
```

Please make sure that any queues you specify in rate tables are created before operators start pricing ads. Otherwise, they'll get a missing file error.

10.4 *Beep Commands*

The ad-entry module can sound three different kinds of speaker tones when pricing an ad. The command form is:

```
Beep [1,2,3]
```

Where the number indicates the type of tone. Using a "1" will result in a short "thud" noise — usually used for keystroke and other minor errors. A "2" will sound the "bring" that the classified system usually sounds when a field value isn't allowed. A "3" will sound the long, ringing noise that's used for major errors like a missing file.

10.5 *Message Commands*

You can make the ad-entry module pop up a message window containing specified text with the Message command:

```
Message "text string"
```

For example, let's say that you wish to prompt operators to check the ad text when a reply box has been ordered. You could add the rate table code:

```
IF BoxType > 0 THEN  
Message "Is box number in ad text?"  
ENDIF
```

Messages may also be multi-line. Let's say you wish to prompt for an up sell when certain criteria are met:

```
IF (InsOrd < 3) & (TextCount < 25) THEN  
Message "ASK:" +  
" " +  
"Would you like to run this ad three times?" +  
"The 3X special for this ad is only 50 cents more."  
ENDIF
```

Enter a multi-line message by putting a "+" symbol at the end of each continued line.

➤ **String Variables**

If you're using the same message in more than one place in a rate table, you may wish to use a string variable to hold the message text. Define a string variable using the command form:

```
String varname "string text" [+]
```

Then, you may use the variable name anywhere in the rate table after the definition point. For example, if you were using the up-sell prompt described above in multiple tables, you could define it as a string variable with the command:

```
String ThreeSell "ASK:" +  
" " +  
"Would you like to run this ad three times?" +  
"The 3X special for this ad is only 50 cents more."
```

Then, just use the code:

```
Message ThreeSell
```

Anywhere you need to send the message. Using string variables like this will save you space in the classified system's string table. That can help performance in low-memory situations.

➤ **Non-Repeating Messages**

If you use the message command described above, the message will be displayed each and every time the ad is priced in the ad-entry module. That will be each time the cursor passes through a price sensitive field after the first pass through the ad form. This could be tedious if lot's of fields are being changed. To avoid this, use the "NRMessage" (*for No-Repeat Message*) rather than "Message" command. This will keep the same message from being displayed more than once while editing a single ad. (*A message is "the same" if it's a repetition of the same command or if the text is based on the same string variable.*)

10.6 *Newer Variable Names*

➤ **Preset Variables for Ad Pricing**

Here are the long-variable names and their one-character equivalents.

Letter	Name	Description
A	Incremental	Acct Charging method 0 full-charge, 1 incremental
B	BoxType	Blind box type 0 none 1 hold 2 forward 3 other
C	Charge	Charge
D	OpenItem	Accounting method 0 balance forward 1 open item
E	Editions	Editions (<i>a set variable</i>)
F	Eds_Sched	Number of editions scheduled
G	UserDef1	ASCII value of user-defined field one
H	UserDef2	ASCII value of user-defined field two
I	InsOrd	Inserts ordered or TFN frequency
J	TFN_Stat	TFN status 0 ad is not TFN 1 ad is TFN
K	Credit	Account credit status 0 OK 1 Overdue 2 No Credit
L	Term	Account term 0 transient 1 long-term 2 external AR
M	InsMade	Inserts made to date
N	Aged0	Account aged balance, portion aged 0X
O	Class	Classification number
P	Aged1	Account aged balance, portion aged 1X
Q	Aged2	Account aged balance, portion aged 2X
R	Table	Rate table number
S	Specials	Specials (a set variable)
T	Seller	Ad seller number
U	Aged3	Account aged balance, portion aged 3X
V	Aged4	Account aged balance, portion aged 4X
W	TextCount	Text count
X	Due	Account balance due
Y	AcctSeller	Account seller number
Za		Number of inserts scheduled in edition "A"
Zb		Number of inserts scheduled in edition "B"
Zc		Number of inserts scheduled in edition "C"
.		
.		
.		
Zz		Number of inserts schedule in edition "Z"
	Scratch	Use as you please

IsNew	Is ad being newly entered	0 No 1 Yes
ChargePerIns	Charge per insertion	
Flags	Ad state flags	1 Ad is skip scheduled 2 Ad entered w/ 4.1+ 3 Sort-key overridden
Module	Program running	1 Ad Entry 2 Auto-Receivables Logging 3 Late Fee/Write-Off
Field	Force Next field	

➤ **Preset Variables For Late Fee Computation**

Letter	Name	Description
A	Incremental	Acct charging method 0 full-charge, 1 incremental
C	Charge	Portion of balance eligible for late charge
D	OpenItem	Accounting method 0 balance forward 1 open item
G	UserDef1	ASCII value of user-defined field one
H	UserDef2	ASCII value of user-defined field two
K	Credit	Account credit status 0 OK 1 Overdue 2 No Credit
L	Term	Account term 0 transient 1 long-term 2 external AR
N	Aged0	Account aged balance, portion aged 0X
P	Aged1	Account aged balance, portion aged 1X
Q	Aged2	Account aged balance, portion aged 2X
U	Aged3	Account aged balance, portion aged 3X
V	Aged4	Account aged balance, portion aged 4X
X	Due	Account balance due
Y	AcctSeller	Account seller number
	Scratch	
	Module	Program running 1 Ad Entry 2 Auto-Receivables Logging 3 Late Fee/Write-Of

11 *DOS Extender*

The ad entry program makes use of memory beyond the normal DOS 640k by using the protected mode available on Intel's '286 and later processors. Protected mode programs may use up to 16 megabytes of properly installed and configured RAM memory.

Although the ad entry program can use extra memory to run faster, it will actually run — though slowly — with less total memory than required by version 4.0.

The programs will only execute on computers with 80286, 80386, 80486 or Pentium processors. Some early '286 machines with older BIOS and chip sets may not be able to reliably enter protected mode. If a particular 286 computer will run Windows 3.1 (*or Windows 3.0 in Standard mode*), it will very likely run the ad-entry program. The program can use memory beyond the normal DOS 640k when it is installed as *extended* memory. Memory that is installed as expanded memory by old-style expanded memory boards is not useful. If you have a 16-bit memory expansion board that's providing expanded memory, you may wish to dig up your old documentation or installation programs to see if it's possible to reconfigure the memory to be extended memory starting at a 10000H address.

11.1 *Memory Managers*

In theory, the newer programs should work with the following memory managers:

HIMEM.SYS
QEMM386
EMM386
386MAX (latest version only)
Windows 3.0 and 3.1
OS/2 2.0 DOS-compatibility box

They should also work with no memory manager, but then must be the only programs using high memory. (*No TSRs may share extended memory.*)

We have tested successfully with QEMM version 6 and 7, EMM386 version 4.2 and 4.44, HIMEM.SYS 2.77 and Windows 3.1.

HIMEM.SYS and EMM386 are included with current versions of DOS. If you're using a 386 or above machine, QEMM386 or EMM386 are good choices since they allow sharing of memory as expanded and extended memory. If you're using a 286-based machine, you'll have to use HIMEM.SYS and may need to specify the machine in the HIMEM command line in Config.Sys.

An included program, XStats.Exe may be used as a diagnostic. It will identify an extended memory manager and report how much memory the extender can use. Run it with a "T" command line flag to actually test the memory.

Context-Sensitive Help and Defaults

3/13/1995

Chapter

2

1 *Introduction*

The SunType Classified System version 4.1 offers customizable, context sensitive help for operators using the account, ad, payment and adjustment entry module. While using that module, an operator may press the **F11** key for general help or the **F12** key for help that is sensitive to the program mode or the form field in which the cursor resides.

Help entries on any given topic may be short or long. The help window pages or scrolls to display multiple screens of help text.

The help contents provided with the classified system is very general and may not accurately describe the way the classified system is used at your publication. The help system is customizable so that you may remedy this deficiency. Examples of ways in which you might wish to customize the help include changing it to:

- **Show a greeting script as the first screen when the general help key, F12, is pressed;**
- **Give detailed help on selecting rate tables or special charges and discounts in accordance with publication policy;**
- **Describe how you want the user-defined fields to be used;**
- **Be specific about the needs of particular zones, since the help contents may be zoned.**

This chapter describes how to customize the help information so that it describes the way the classified system

1.1 *Field Defaults*

The help system also provides a mechanism for setting the defaults for several fields in the account, ad and detail forms. This default system supplements the classified system's other mechanisms for setting defaults — for example through the “Set General System Characteristics” module — in the following ways:

- **It allows you to set defaults for some fields that may not be set any other way;**

- **Since the help/default system may be zone sensitive, it allows you to customize defaults by zone;**
- **It allows you to set limits on use of the user-defined fields, providing some needed input restrictions.**

12 *Editing Help*

If the above seems tantalizing, you'd probably like to get right down to customizing your own help. To do so, you use a text editor to edit an ASCII text file named HelpTbl.Fil. To edit this file and compile the results, run the "Set General System Characteristics" module from the Installation and Maintenance menu and choose item "9" from the list on the second screen. This runs a text editor to edit the correct file and, when you exit the editor, will compile the help for rapid, context-sensitive help lookup.

2 *Help File Structure*

The help compiler reads a single ASCII text file contains the help text plus markers for the title, context segmentation and bold lines. The header is a two or three line header that's always displayed at the top of the help window. The context segmentation markers tell the system where the help on particular contexts resides. Finally, lines may be marked to display in bold, usually for section titles or emphasis.

2.1 *Title*

The help file title is always displayed at the top of the help window. In the help text file, it's the first few lines. It's end is marked by a line containing only an equal sign "=".

For example, the title area as shipped appears:

<p>Classified Help =</p>

If you're zoning your help and wish a publication name to appear as a help title, you could easily change this to read something like:

<p>The Dune Observer (Zone 035) =</p>

Keep the title short: every title line means one less visible line of on-screen help.

2.2 *Segmentation*

The help file needs to be segmented into contexts. Each context is defined by a "key word" that the help compiler knows to stand for a particular help context. This key

word should appear, preceded by a colon, on a line by itself. For example, the key context. This key word should appear, preceded by a colon, on a line by itself. For example, the key word for the context for the last name field on the account form is “LName”. So, the line setting the context for help for the last name field might appear:

```
:LName  
Here's the help for the last name field ...
```

The same help text may be assigned to multiple contexts. For example, you may wish to write one set of text for the address lines on the account form and have that text serve as the help for all three lines. To do so, you would set the contexts like this:

```
Address1  
:Address2  
:Address3  
Here's the help for all three address  
fields...
```

:

23 *Bold Lines*

The help system will embolden any line that begins with a caret “^”. This is most useful for marking subtitles. For example, the help for the last name field actually begins with:

```
:LName  
^Last Name  
The Last Name Field: What's worth noting  
about this field is that the first ten  
characters entered here will be a ...
```

When displayed in the help window, the line reading “Last Name” will be displayed with an emphasizing video effect.

24 *Paragraph Formatting*

The help compiler automatically reforms lines to fit within the help window. That means, though, that it has to be able to tell the difference between a line break and a paragraph break. The method for telling the two apart is simple: if a line ends in a space, a hyphen or a slash, it's regarded as a line break. If it ends with anything else or is an empty line, it's regarded as a paragraph break. The help compiler will join lines and word wrap as necessary between paragraphs, but will not join paragraphs.

Lines in the help file may be as long as you like so long as there are word breaks at least every 50 characters.

The total length of the help file is currently limited to around 65,000 characters.

3 *Help Contexts*

Key	Word Context
Searching	
BrowseQueue	Queue browsing
SearchPhone	Searching by phone#
SearchName	Searching by name
SearchAccount	Searching by acct#
BrowseName	Browsing by name
BrowsePhone	Browsing by phone#
BrowseAds	Browsing ads
SearchAround	Search fwd/back, accept
Account Form Fields	
FName	First name field
LName	Last name field
Address1	Address line 1
Address2	Address line 2
Address3	Address line 3
Phone	Phone number
Type	Account term
Memo	Account memo
UserDef1	User-defined 1
UserDef2	User-defined 2
Credit	Credit status
Method	Bookkeeping method
Posting	Incremental/Lump Sum
Seller	Account seller
AbortAccount	Abort query
FieldSelAcct	Account correction mode
Account Summary	
DetailSummary	Account summary screen
Ad Form Fields	
Zone	Ad zone
AdSeller	Ad seller #
Start	Start date
InsOrd	Inserts ordered
Editions	Editions
Skip	Skip scheduling

Class	Classification #
Table	Rate table
Specials	Special charges and discounts
MailBox	Reply box field
AdText	Text editor
AdNote	Ad note field
VoiceBox	Voice box editing
SortKey	Sort order spec.
Count	Text count
InsToDate	Inserts to date
Charge	Ad charge
FieldSelAd	Ad correction mode
Change	Charge Query when charge changes
ChangeExplain	Request for change explanation
AbortAd	Abort query
Duplicate	Duplicate query

Classified Display Fields

DisplayTag	Tag
DisplayCols	Column width
DisplayDepth	Ad depth
DisplayInLine	InLine?
DisplaySource	Source file
DisplayPrint	Print file

Payments & Adjustments Fields

DetAmount	Amount of payment/adjustment
DetDate	Entry date
DetSeller	Seller #
Invoice	Invoice# query for open-item
DetType	Detail type (cash, check...)
CashNote	Note for cash
CheckNo	Check number
CheckNote	Note for checks
CardType	Charge card type (MC, Visa...)
CardNo	Charge card number
ExpDate	Charge card exp date
AuthCode	Charge card auth. code
CreditType	Other credit type
CreditNote	Other credit note
ChargeType	Other charge type
ChargeNote	Other charge note
DetNote	Detail memo item
FieldSelDetail	Detail correction mode

AbortDetail Abort query

Miscellaneous Contexts

AddQueue Adding item to queue

4 Defaults

The help system may be used to set defaults for many fields on the account, ad and detail forms. To set a default for an eligible field, you generally should place the value you want to use on the context line with a space separating it from the context key word. For example, to set the account memo field to start with the contents “New Account” for new accounts, you’d write the context line as:

```
:Memo New Account  
Here’s the memo field help...
```

And to set an initial value of “550” for the classification field, you’d use the line:

```
:Class 550  
Here’s the classification field help.
```

The mechanism for the account form’s two user-defined fields is a little different because it allows you to specify both the default and the set of characters that will be considered valid. Let’s say you want to limit the input for the first user-defined field to “Y” or “N” and set the default to “Y”. The help file entry would look like:

```
:UserDefl Y YN  
Here’s the help for the field...
```

The first “Y” sets the default. Then, the “YN” limits input to the characters “Y” or “N”. Case is not significant.

The default mechanism for the ad start date is also different. It may be used to set the number of days that should be added to, or subtracted from, the current date to pick a default start date for new ads. For example, if the start line in the help file looks like:

```
:Start 7
```

The, when a new ad is entered, the program will take the current date, add seven days, then search for the next available publication date on or after the sum date. This facility is particularly useful when using the classified system’s zone option with zoned editions since the “next publication date” entered when starting the ad-entry module is meaningless in this context.

Following is a list of context key words that may be used with default settings:

Keyword	Argument
Account Defaults	
Start	date offset
Address1	string
Address2	string
Address3	string
Type	T L E
Memo	string
UserDef1	character & set
UserDef2	character & set
Credit	A B C
Method	O B
Posting	I N
 Ad Defaults	
InsOrd	0-999
Editions	string of A-Z
Class	1-9999
Table	1-99
Specials	string of A-P
AdNote	string
 Payment/Adjustment Defaults	
DetType	A B C D E F
CashNote	string
CheckNo	string
CheckNote	string
CreditNote	string
ChargeNote	string
DetNote	string

Selection-Time Ad Sequence Scrambling

Chapter

3

Classified System Technical Bulletin: Selection-Time Ad Sequence and Scrambling

1 Introduction

Facilities newly added to the 4.1 version of the typesetting selection module allow edition-specific scrambling of ad sequence at the time of typesetting selection.

This facility was developed for a customer with multiple products who wished to produce some editions with scrambled ad sequence and some without. Also, each edition might have some classifications that would be scrambled together while others would be produced in classification order. And, the customer needed to be able to track ads by classification even when scrambled.

2 The Scramble File

The selective scrambling facility may be turned on by creating an ASCII text file, Scramble.Fil, in the classified *data* directory. This file should contain a list of editions to be scrambled and the classifications that should be excluded from scrambling. The Scramble.Fil contents should look like:

```
editions abc
except 1,37
edition de
except 1,50,51
```

This would specify that editions a, b, c, d and e should be scrambled at typesetting selection. For editions a, b and c, classifications 1 and 37 should be excluded from the scrambling (meaning that ads in these classifications will print sorted under their class titles as usual). For editions d and e, classes 1, 50 and 51 should be excluded from randomization.

The scramble file may contain as many edition/except pairs as necessary. Separate each pair from others with an empty line. Excepted classifications will print at the start of the ad run, scrambled ads at the end. All scrambled ads will be printed under the header for class 9999.

Editions not listed in the scramble file will not be scrambled.

3 *How it Works*

When typesetting selection is done, the classified system writes out a disk file (Current.Fil) containing a packed listing of the ads selected for typesetting. The typesetting module (choice “B” on the typesetting menu) reads this file when it typesets ads (or prepares ClassLink files...).

This version of the typesetting selection module looks for Scramble. Fil as soon as typesetting selection criteria are entered. If it finds the file, it looks for the current edition in the scramble file list. If it doesn’t find the file or doesn’t find the edition letter listed in the file, ads selection proceeds as usual. If the scramble file is found and the current edition is specified in the file, the scrambling mechanism is activated.

If scrambling is activated, the order of ads in the current file listing is randomized except for ads placed under the excluded classes. The excepted classification ads are listed at the start of the list, the randomized ads at the end under class 9999.

The order of ads in the classified database is not changed.

4 *Notes on Reporting*

The breakdown of selected ads produced by the typesetting selection module is unchanged by the scrambling facility.

If the typesetting selection analysis module (choice “D” on the typesetting module) is used to generate reports on selected ads, you should keep in mind that the reports are based on the Current.Fil. That means that if no sort criteria are specified, the ads will be reported in production order. If you need a breakdown by the class numbers under which the ads were entered, you should specify a sort by class.

Index

A

Account Details 1-2
Ad & Account Browsers 1-1
Ad Indexing 1-11
Ad Order Change 1-4
Ad Scheduling 1-5
Ad Text Editing 1-2

B

Back Charging 1-23
Bad Debt Write Off 1-3, 1-9
Beep Commands 1-31
Bold Line 2-3
Box Shading 1-20
Brower Options 1-11

C

Commands Messages 1-31
 Beep 1-31
Context Sensitive Help 1-1
Contract Tracking 1-22,23
Credit Card Memory 1-22

D

Dos Extender 1-7, 1-
Defaults 2-6
Delete Ad & Detail 1-9
Display Suppressing 1-22

E

Edition Scheduling 1-22
Editing Exceptions 1-14
Editing Help 2-2
Extended Memory 1-7
Expired Ad Locking 1-22

F

File Backup 1-3
Field Defaults 2-1
File Manger 1-27
File Structure Help 2-2

G

General Queues 1-1
H&J's 1-20
Help 1-1, 2-4
Hyphenation 1-6
 Controlling 1-13
 Dual Language 1-6

Language 1-14
Option 1-12

I

Incremental Charge 1-23
InLine Display 3-4
Insertion Changes 1-24
Install ClassLink 1-2

J

Justified
 Columns 2-6
 Vertical 2-6

K

Key Lock 1-22

L

Late Fees 1-26
Leading Auto 1-19
 Leading Alternative 1-19
List By Name 1-5
Live Ad Indexing 1-2
Lockouts 1-10
Logos Display Handling 1-2
Logos in Text Ads 1-14

M

Mail Box Templates 1-21
Memory Managers 1-35
Message Commands 1-31
Misc. Options 1-21
Modularize 2-9

N

Name Variables 1-27
Negative Tracking 1-20

O

Over Ride Flags 1-3, 1-28
Overstrike Mode 1-25

P

Password Protected 1-3
Paragraph Formatting 2-3
Permission Controls 1-3
Printer Specifications 1-8
Production Notes 1-16

Production Sort 1-26
Production Time Ad 1-4

Q

Query End Program 1-25
Queues 1, 1
 General Purpose 1-1
 Rates 1-3
 Rates 1-3

R

Rate Table 1-28
Reporting Notes 3-2

S

Scramble File 1-2
Scripting Option 1- 3
Scripting 1- 8
Segmentation 2-2
SunType 4.1 Release 1-1
SetFonts 1-27
Setup V4.1 1-7
SMSC Programs 1-8
Spelling 1-6

T

Time Ad Sequence 3-1
Title 2-2
Typesetting 1- 5, 3-2
Typesetting Format 1-5
Typographic 1-5
 Condense/Expand 1-18
 Scaling 1-16

U

Utility Editor 1-25

V

Variables 1-29
 Ad Pricing 1-32
 Charge Per Inch 1-29
 Fields 1-29
 Field Numbers 1-30
 Module 1-29
 Non-Repeat 1-32
 Queue Addition 1-30
 String 1-32
Voice Box Templates 1-21

W

Warranty I
Windows Fonts 1-5
Word Counting Opt. 1-24
Write Off 1-26