

Volume

3.1

THE CLASSIFIED GROUP, INC.

SunType Classified Advertising Publishing Systems



Operation manual

Version 4.3 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.
DBA/SunType Publishing Systems
PO Box 512108
Punta Gorda, FL 33951-2108
800-458-4605
Fax: 941-876-6071

[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 © 2001-2005 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	
Installation Instructions	
Installing 4.3 as upgrade to 4.2	1
Adobe Acrobat Reader Online	2
Installing Adobe Acrobat Reader	3
CHAPTER 2	
New Features of V4.3	
Fax & Email Account Information	4
Sample Menu Picture	4
Report Generator Changes	5
NoDetail	5
New Invoice Variables	6
Re-aging Account Balance Utility	7
Sample Re-aging Menu	8
STImport Utility	9
Sample Menus	10
Special Character Translation	11
Multiple Counters	12
Variable Counting	12
Changing Flag in Ad Entry	13
Miscellaneous New 4.3 Features	14
Requiring Input in User-Defined Fields	14
New Ad Duplication Option	14
Ad Duplication & Seller #'s	14
Command Line Proofing	15
Command Line flags & Arguments	16
Proofing by Voice Box	17
Spell Check	18
Year Numeric Sort	18
Selective Ad Purging	18
On Line Credit Interface	18
CallChecker (Do Not Call List)	19

Pagination 19

CHAPTER 3	
HTML Support Package	
HTML Support Package	1
HTML Features	1
Program Components	2
Other Components	2
Setup	3
HTML Dat.File	3
Class Header Styles	4
Classification & Ad Breaks	4
Ad Page Headers & Footers	5
Previous, Index & Next Markers	5
Index Page Headers	6
Class & Subclass Markers	6
Character/Images Substitution	7
Interleaving House Ads	7
Netscape Extensions	8
Unique Ad Id's	9
File Name Conversions	9
Generating HTML Ad Run	10
WWW Posting	10
Sample HTML Data File	11
Special Character Translations	13

INDEX

1.1 *Installing Version 4.3 as Upgrade from 4.2*

Before proceeding with any upgrade, **you should make a current backup** of all SunType program and data files. The following procedure creates an additional copy of the files during the upgrade process. If you wish, you may rely on this procedure as the required backup.

These instructions assume the use of the “standard” SunType directory structure for 4.2, which is a folder named “SunType” containing folders named “Clprogs” and “Cldata”. If you are using different folder names, substitute them as appropriate in the following instructions. It might be a good time to convert to the “standard” folders if you wish.

NOTE: Upgrading to Version 4.3 from 4.1 or earlier will require that you follow the instructions for upgrading to 4.2 before attempting to upgrade to 4.3. (See the 4.2 Release Manual for install instructions)

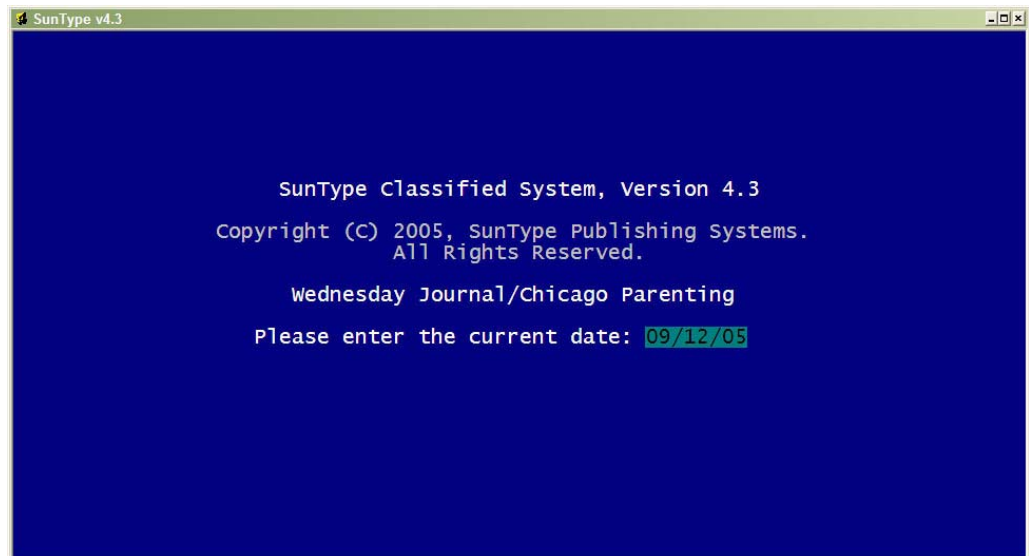
After making the require backup:

1. Open the SunType folder on your server or local hard drive.
2. Make a copy of the Cldata folder and paste the copy into the same SunType folder.
3. Rename “Clprogs” to “Clprogs.42”. Rename “Copy of Cldata” to “Cldata.42”.
4. Make a new “Clprogs” folder. Open the “Clprogs.43” folder on the CD (early CD’s named it “v4.3b”).
5. Select all of the files and copy them to the empty Clprogs folder on the hard drive.
6. Locate the “42to43.bat” file in the Clprogs folder and double-click it. It will run the data through a brief conversion process to bring the data up to 4.3 standards.
7. *Note:* If you are *not* using the “standard” SunType” folder names, or are running versions of Btrieve called “Brequest” or that require “Btrbox95.Exe” (Windows 98), you’ll need to edit the 42to43.bat file before proceeding to change the “stf files” setting and/or the version of

Btrieve. If you are running SunType's original Btrieve or Pervasive SQL on Win 2000 or XP, you don't have to change the Btrieve settings. If you have problems in this area, call for advise.

8. Locate the batch file you've been using to start SunType in the "Clprogs.42" folder and move it to the new Clprogs folder. It's usually called "Runclass.bat". If it isn't, you can determine the name of the file you're looking for by checking the Properties on your desktop shortcut.
9. There is a new SunType icon in version 4.3 named "Stwin.ico" which you may apply to your shortcut if you wish.

When you run SunType for the first time after upgrade, verify that the sign-in and Main Menu screens display Version 4.3 on the screen.



(SunType Version 4.3 Sign-in Menu)

1.2 *Adobe Acrobat Reader-On-line Documentation*

Your upgrade or program set includes, in most cases, copies of much of the SunType documentation in Adobe Acrobat format (PDF). The full 4.0 manual is included, along with supplementary documentation for 4.1 - 4.3. In addition, many of the miscellaneous technical bulletins are also included on your install set. You can find the main manuals at our website www.classifiedgroup.com. The Acrobat Reader installation set is included on your CD.

1.2.1 *Installing Adobe Acrobat Reader:*

You may install the Acrobat Reader (or upgrade an existing installation to version 7.0) by opening the Adobe Acrobat Reader 7.0 folder on the CD and double-clicking “Acrd4Enu.exe” and following the instructions on the screen. This software is subject to the included license agreement with Adobe Systems and is distributed under a license from Adobe Systems. There is no charge for this software license. You may download it from www.adobe.com .

New Features of 4.3

Chapter

2

2 *FAX and Email Account Information*

The account record in the new version has fields for storing an account email address and a FAX number for the account. These fields can be included in all user-defined reports, and can be included in the text of ads in the same way as other account fields are accessed.

Due to space limitations in the current screen layout, the Email and FAX fields are not displayed when the account is initially opened on screen. Future modifications of the screen layout are planned, but the basic screen structure is common to several of the data entry screens, so significant reworking is involved.

Entering/Changing Email and FAX:

To display/modify FAX and Email information on existing accounts, choose option "C" to edit the account information while the customer record is displayed on screen.

During new account entry, both fields are visible and accessible while the account is being set up.

Using Email and FAX in Ad Entry:

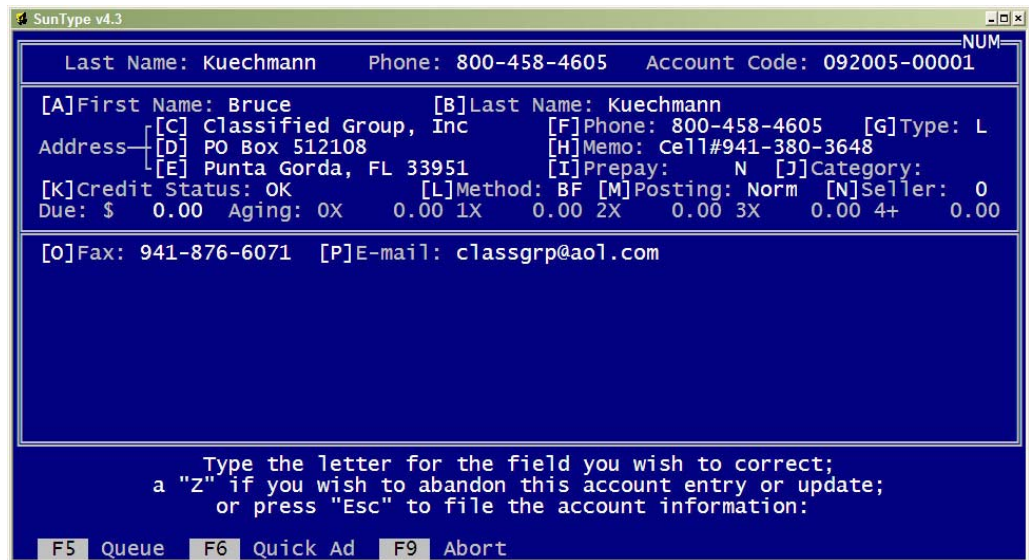
The Email address and FAX number may be inserted directly into classified ad text using the following keys:

<ALT>-M inserts the Email address.

<ALT>-N inserts the FAX number

These keys are listed, along with other insert options, on the <F8> menu during ad text entry.

Information on using these fields in reports is covered in the section on the Report Generator below.



(SunType's New Fields)

2.1 *Report Generator Changes*

Instructions on the use of the Master Report Generator are contained in Appendix A of the SunType manual, and on the release CD as "Repgen.pdf" in the Stdocs folder. The pdf version is the most current available, so it's generally the one to use.

FAX and Email Fields:

The new FAX and Email fields in the account records can be reported using the following variables:

Email String Email address field from account

FAX String FAX number field from account record

Account-based variables are available to all sections of the report generator, so these fields can be called from any report module.

2.2 *NoDetail*

Prior to this release, there was no way to extract inactive account records into a list, queue or other report – a prudent step prior to running the "Purge Inactive Accounts" procedure from the Maintenance and Error Recover menu.

Obsolete ad and detail records are purged during the aging process based on user-specified cutoff dates. Once all detail has been removed from an account

record by the aging process, indicating no recent activity, it is deemed “inactive” and eligible for purge.

The “Purge Inactive Accounts” process on the Maintenance and Error Recovery Menu allows separate purging of Transient, Long Term and External account types – the intended usage is to change Long Term and External accounts that are to be purged to Transient status to prevent unintended loss of account information.

In this release, a list or queue of inactive accounts can be created for review by running an account-based report with “NoDetail = 1” in the Selection Criteria.

NoDetail is a Boolean variable which returns 1 if true and 0 if false. So, if an account has no remaining detail, it will return a value of 1.

For example, the following line in a report would return a list of inactive Long Term accounts:

```
.SELECT (NoDetail=1)&(Term=1)
```

Like other “lists”, inactive accounts may be sent to a queue for on-screen review prior to running the purge function. If an account is to be retained, a Memo detail item can be entered to remove it from inactive status.

A basic sample of an inactive account report is included on the 4.3 release CD in the Etc folder. Copy it to the Cldata folder and look for it in the Account-based report list in the Master Report Generator.

23 *New Invoice Variables*

SunType’s Master Report Generator includes several variables that can be used to report or analyze individual invoice records on Open Item accounts. The existing variables include Icount, Idue, Icredit, Idebit and Ikind. These are documented in Appendix A of the manual or in the RepGen.pdf included in the Std docs folder on the release CD. (Also See *updated Report Generator PDF*)

To allow the association of payment and adjustment transactions with specific editions or combinations, the following new variables have been added:

IEditions	Set	Used for testing or selecting the edition(s) included in a specific insertion order-based invoice.
------------------	------------	---

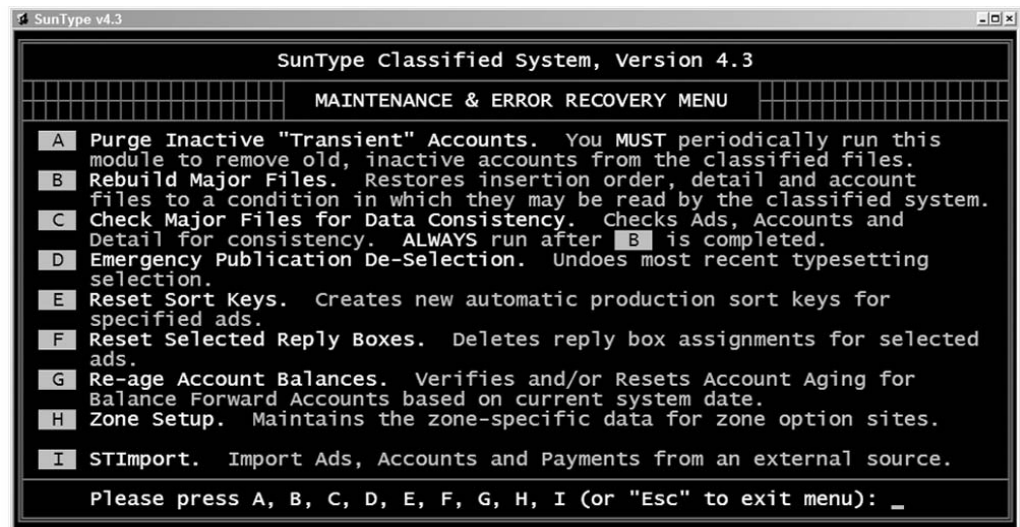
IEditionsList String (26) Used to print list of included editions.

In an Open Item system, the lead transaction on most reference documents (invoices) is an insertion order. In fact, it's not possible to add an insertion order to an existing invoice. Payments and adjustments may be entered with their own invoice numbers – the Late Fee and Write-off Small Balances function in the Billing and Aging section of the Main Menu distributes unapplied credits to unpaid invoices with balances.

If you are using edition-specific analysis with these variables, “on account” payments and credit memos will not be reported until they have been applied to open invoices containing an insertion order. You should avoid using “on account” debit memos or miscellaneous charges, since they will not have an insertion order associated with them and, like an insertion order, may be paid by automatic application of a credit to them.

2.4 *Re-age Account Balances Utility*

This utility was formerly available as a stand-alone function, but has now been added to the Maintenance and Error Recovery Menu (G) for easier access. It can be used to verify and correct anomalies in aging information, or, more commonly, to undo the effects of accidentally running the Account Aging module more than one time at the end of a month.



(New Maintenance & Error Recovery Menu)

Use this utility with caution, and be sure you have a current backup, until you are comfortable with its operation. However, since it doesn't purge any data the way the regular aging process does, it can be re-run to correct any mistakes.

It operates by re-computing all balances and aging information based on the actual detail stored in the account. All data is considered, whether posted or unposted, when determining the aging status of the account.

This utility is *not applicable* to systems running all accounts on an open-item basis. Open item accounting is specific to the date of the invoice rather than simply using monthly "buckets" as in balance forward accounting.

The effect of this utility is controlled by the system date that you confirm when starting the program. Choosing the appropriate system date is essential to obtaining desired results.

Based on the system date, cutoff dates will be computed for each aging period according to the number of days stipulated in the General Characteristics section of Installation Operations as the billing and aging period. It's normally 30 days, but other intervals such as 7 and 14 days are also possible.

When it is run, the program will treat all transactions occurring in the most recent aging period, as well as any beyond the current date, as "0X", or Current. Each previous aging interval goes in turn into "1X", "2X", "3X" and "4X", indicating the number of times each item has been aged.

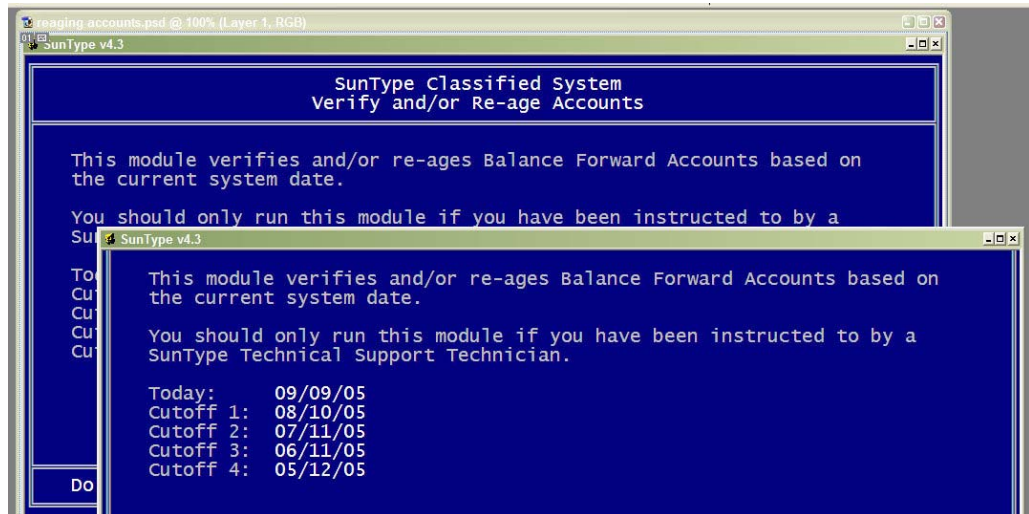
The cutoff dates the program will use based on the current system date are displayed prior to beginning the process for your confirmation. Accounts for which changes are made are placed in a queue called "Reaged".

Here's an example of how the choice of system date will affect the results of this program:

If today's date is June 1 and the system date is also June 1, the resulting data will show all May activity as "0X" (Current), as though the May aging has not yet been run. June activity will also be current.

If today's date is June 1 and you set the system date to July 1, the result will show all May activity as "1X" (30 days) and any June activity will be in "0X" (Current). *Be sure you exit and reset the system date prior to doing other work on your data!*

A PDF called Reag42.pdf is included in the Stdocs folder on your release CD. It contains a somewhat more detailed discussion of the re-aging utility.



(Sample Re-Aging Menu Utility)

2.5 *STImport Utility*

STImport is a utility that facilitates importing of account information, ads and payments into SunType Classified for approval and publication. Previously available as a stand-alone program, it can now be run from Maintenance and Error Recovery Menu in SunType.

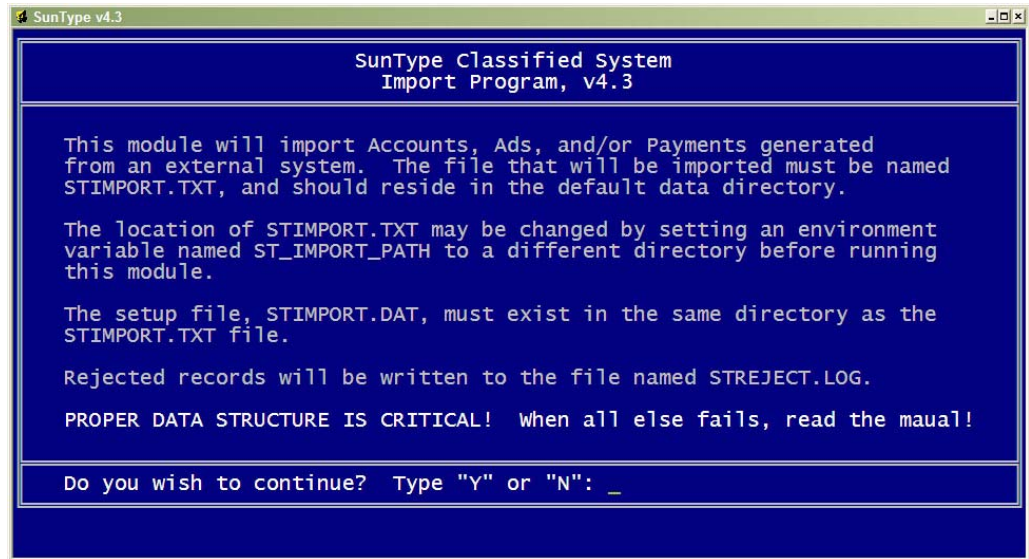
STImport is typically used to import ads from outside sources, such as web sites, outside databases or other classified systems. It can also be used for Scan ads or any other source provided the ads are put in the proper format.

STImport is used when repetitive imports of ads from specific sources are anticipated, since it does require some setup and testing before use.

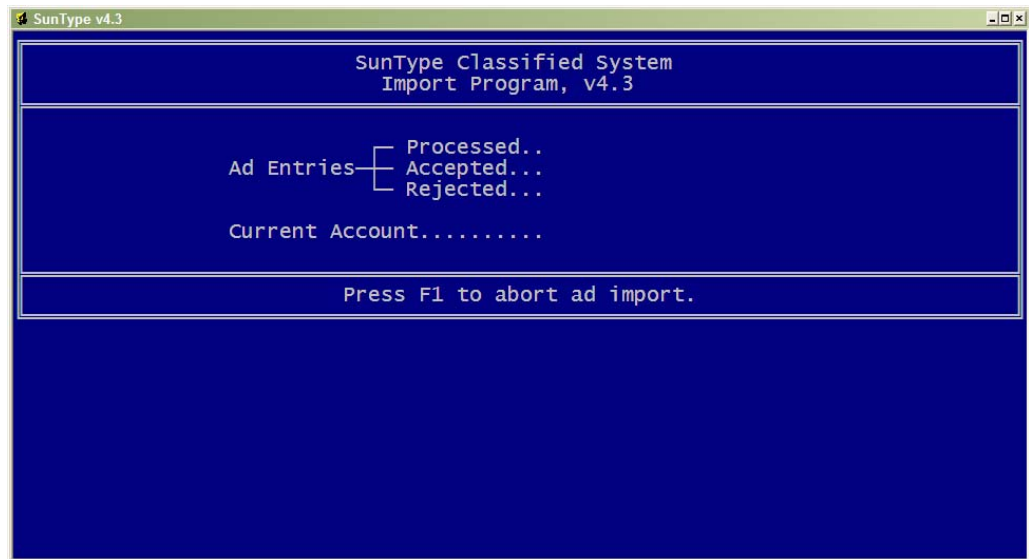
As an example, a SunType customer has used STImport to bring in 1200-1500 new rental listings from an agency on deadline each week. The entire processing of the ads for publication requires about 5 minutes to complete.

When used with a web site, the information entered on the web is sent to SunType with the appropriate tags attached to each part of the ad transaction. New accounts are set up as needed, ads are entered into the system and credit card payments, when appropriate, are brought in with the ad copy. All imported data can be queued up for review prior to processing the payment and releasing the ad for publication.

There are a number of components to the initial setup, so detailed information must be obtained from STImp43.pdf, which is included in the stdocs folder on the release CD.



STImport Menu Sample



(Sample STImport Run)

2.6 *Special Character Translations*

There are a number of characters that cannot be typeset from SunType using direct keyboard input because those characters are reserved for system functions. Prior to the advent of the Web, these keys had little typesetting impact (i.e., the tilde (~) character was rarely used as a stand-alone object prior to the Internet because it was normally combined with another character, as in “ñ”).

SunType solves this problem by substituting other unused character values during ad entry and then translating them to the desired typeset character on output.

Some have been around for quite a while, and a few additional characters have been added lately to allow embedding of HREF codes into ads to create live links in ads sent to the Web.

Here is the list:

Character Desired	ASCII value	Win Char*
<	231	ς
\	221	Ÿ
^	226	â
{	222	□
	223	□
}	230	μ
~	227	π
>	232	□
"	233	□

*These characters don't necessarily correspond to what you would get in SunType, which uses a different character complement when editing. Most will produce a Greek character of some kind.

You may choose to employ these characters as needed in SunType macros – this approach is necessary when working with HREF formatting strings. “ASCII” values are entered from the keyboard by holding the <ALT> key down and typing the number on the *numeric keypad*, not on the regular keyboard.

When doing HREF coding for the web, the macros for the print output are set up to use Font 98 {f98}, which is the non-printing (hidden) font in SunType. Examples of this setup are available from SunType support.

2.7 *Multiple Counters*

Classified versions 4.1 and earlier had one “counter” that accumulated the word, character or typeset measure of an ad. Most rates were written by checking the value of the rate table “W” or “Count” variable.

There are now five auxiliary counters. You may determine which is accumulating line or typeset measure counts as the ad is priced through typesetting format directives. For example, if an ad text read like this:

```
{%1}Now is the time.  
{%2}For all good persons to come  
to the aid of the quick brown fox that  
jumped over the lazy dog.  
{%3}Call Jan, 123-4567.
```

The “%” typesetting format command has been used to divide the ad into three regions. When the ad is priced, the rate table variable “MCOUNT1” will hold the measure of the first line; “MCOUNT2” will hold the measure of the body of the ad; and “MCOUNT3” will hold the measure of the last line (and any rate-table off code).

The old “Count” or “W” variable will contain the full count of the ad.

Note that use of the “%” format command in an actual ad would probably be unusual. It would be more typical to hide it in a typesetting macro so that the salesperson doesn’t have to know about it.

The new auxiliary counters are named MCOUNT1, MCOUNT2, MCOUNT3, MCOUNT4 and MCOUNT5 for use in both rate tables and the report generator.

2.8 *Variable Counting*

Versions 4.1 and earlier had problems pricing by typeset lines if the charge for each line wasn’t proportional to line spacing. A common workaround to this problem was to set line spacing commands in the screen formatting command table to proportional to the desired charge for each type style, then use real line-spacing commands in a separate output table. This workaround fouled the typeset measure variable stored with each ad, making it impossible to get good reports on ad run length.

Version 4.3 solves this problem by allowing you to set — via typesetting command — the value to be accumulated for a line. These values may be saved in the new auxiliary counter variables.

```
{ $3 } This is the headline.  
{ $1 } This is the body of the ad, running two lines.
```

For example, if an ad is coded:

Then, on pricing, the MCOUNT1 variable would hold a value of five, three for the single headline plus two lines at one each.

MCOUNT1 is used by default for variable pricing — but you may use the “%” counter set command to use any of the five auxiliary counters.

Let’s look at another example of variable-line pricing. Here’s a sample ad:

```
{ $450 } This is the headline.  
{ $125 } This is the body of the ad, running two lines.
```

And, here’s a line in a rate table:

```
Charge = (MCount1 / 100) * Inserts
```

The charge for the ad is now \$7.00: \$4.50 for the headline and \$1.25 each for the two bodylines.

A note on the limits of this scheme: each of the MCOUNT# variable has a maximum of 65,535. Keep that in mind when designing the relation between your ad coding and rates.

2.9 *Change Flag in Ad Entry*

It’s now possible to test in the rate table whether an ad has been changed during the current edit. This only applies when a rate table is evaluated in the ad-entry module. When an ad is new or has been changed during the current edit, ad-order flag eight will be set. For example:

```
IF (NOT IsNew) and (8 IN Flags) THEN  
  Queue “CHANGES”  
ENDIF
```

Queues an ad to the changes queue if it's not a new ad, but has been changed during the current edit session.

Please note: the change flag is not sensitive to ad-text changes that do not affect the first line, text counts or sort order of an ad.

3 *Miscellaneous New 4.3 Features*

- Zones beyond 255 are accepted.
- Ad-Based Reports: use the key CLASS### where “###” is a zone number to limit the report to all the ads in a single zone.
- Proofs: Proofs are generally limited to one zone unless you override the zone in the final dialog.
- Command Line Proofing

3.1 *Requiring Input in User-Defined Field*

4.3 allows you to require the operator set either or both user-defined fields. The new facility is a simple extension of the one that allows you to set defaults and valid responses for user-defined fields via the context-sensitive help definition file.

To set a default and valid character set for a user-defined field, you would ordinarily have a line like:

```
:UserDef1 Y YN
```

in the help specification for a zone. This sets a default of “Y” and limits input to “Y” or “N”. If, instead, you specify:

```
:UserDef1 _ YN
```

Then, the field will start blank. And, when the operator asks to file the account, the field will be checked to see if “Y” or “N” has been specified. If not, a warning box will pop up and the cursor will move to the user-defined field. Only by using the abort entry or change command will the operator be allowed to leave the account-editing context without setting “Y” or “N” in the user-defined field.

(Note: user-defined fields are still named via the “Set General System Characteristics” module.)

3.2 *New Ad Duplication Option*

The ad-entry module's Copy Ad command assumes that what you're trying to do is get a second copy of a live ad. Sometimes that's not the real goal; often you

want to stop an ad and restart a copy of it as a new ad. A new copying option makes it possible to do this with a single command. The command stops the original ad by making its inserts-ordered count equal to inserts made; and, it creates a fresh, new copy of the ad, starting on the next publication date with the original editions and number of inserts ordered. Also, if the original ad has voice or mailboxes, these are cleared from the original ad and copied into the fresh, new ad. So, after executing the command, there will still only be a single ad with the box data and it will be the new, rescheduled ad.

To use the new “renew” command, press the “W” key where you would have used the “X” key, at the “Type the letter for the field you wish to edit” prompt.

If you like, you can set up the ad-entry module so that a “W RenuAd” prompt appears in place of the “X CopyAd” help line at the bottom of the screen. This will only affect editing of existing ads, since it doesn’t really make sense to “Renew” a brand-new ad.

To set up the new help line, add the line:

```
ShowStopDup=TRUE
```

to the ClassAd.Ini file.

3.3 *Ad Duplication and Seller Numbers*

The ad duplication and renewal options normally copy the operator (salesperson) number associated with the original ad to the new ad. You may change that behavior, so that the current operator number is affixed to the new ad, by adding the line:

```
CopyOpNum=FALSE  
to your ClassAd.Ini file.
```

4 *Command-Line Proofing*

This newly added feature of the classified system’s proofing module allows operation of that module from the DOS command line without operator intervention. A typical command line use might be a batch file with the contents:

```
set stfiles= f:\suntype\classad  
btrieve  
proof /d 0 /a=D /T=2 /O=f:\ads\adrun.ps  
btunload
```

The first line tells the proof program where to look for data files. The second line loads the Btrieve file manager (*you may need to use BRequest instead*) and the fourth line unloads it. The third line runs a proof of all ads scheduled for the next edition (/A=D), using the current system date (/D), running as operator #0 (0), using output code table 2 (/T=2), and sending the output to the file f:\ads\adrn.ps (/F= f:\ads\adrn.ps).

Three flags (/A, /D and the operator number) must be present on the command line for uninterrupted batch operation. If any of them is missing, the program will stop and request user input.

4.1 *Command Line Flags and Arguments*

/T=#
Output table number
/O=[file|device]
Output file or device
/S=#
Starting classification number
/E=#
Ending classification number
/A=[A|B|C|D|E]
Ad-selection method.
A all ads
B active (unexpired ads)
C ads entered during date range
D ads due for insert in next active edition
V ads specified in a text file by voice box (see below)
/M=[P|C|R|G|H|M]
Output method (driver)

P PostScript
C ClassLink (XPress)
R RTF
G Generic typesetter
H HTML
M MCS Slave
/F=mo/dy/yr
First date if ad-selection method 'C' is in use.
/L=mo/dy/yr
Last date if ad-selection method 'C' is in use.
/P=mo/dy/yr

Publication date if ad-selection method 'D' is in use. Defaults to next valid publishing date on or after

current date.

/Q= [A . . . Q]

Publication edition. Defaults to first edition on or after current date.

/D

Use current system date (don't ask for a date)

/D=MO/DY/YR

Use date mo/dy/yr

#

Run as ad-taker #. Don't ask for number.

Z=####

Load information for zone "####". Not available unless the zone system is in use.

4.2 *Proofing by Voice Box*

If you specify "/A=V" on the command line, the program will look for a file named "VBType.Txt" in the classified file directory and read it for a list of voice box numbers and zones. Ads specified via voice boxes are read, sorted by the production sort specification and output.

VBType.Txt should be a text file with one line per ad. The line should be

Zone,VoiceBoxNumber

formatted like:

So, for example, the entry for an ad in zone 47 with voice box number 10234

47,10234

would read:

5. *Miscellaneous New Options*

5.1 *Spell Check*

You may now disable the option that allows users to put words into the user dictionary. To do so, add the line:

```
AuxAdd=FALSE
```

to the ClassAd.Ini file. This will cause the “Put in user dictionary” option to disappear from spell-check menus.

5.2 *Year-Numeric Sort*

A new sort order was added (01/07/00) for a non-bug Y2K problem: Some sites were using Numeric sort order with two-digit dates, typically on car ads.

A new sort order “YrNumeric” may be used to fix the problem while maintaining existing ad formats.

For classifications using the YrNumeric sort order, the program finds the first consecutive set of digits in the first ten chars of ad text and converts to a number. If that number is less than 100, 1900 is added if it is greater than 30, 2000 if less than or equal to 30. Then, the number is inverted for the latest year first sort. Affected users will need to specify the new desired sort then run “Reset SortKeys” module to fix old sort key.

5.3 *Selective Ad Purging*

Normally, the purging module deletes old detail based on its date entered, so long as the item (or the lead item in an invoice) is not an active ad. Since ads may run for long periods, this could result in their being purged too rapidly after expiration.

Add the line

```
PurgeOnStop=TRUE
```

to the ClassAd.INI file to force the program to check the ad end date (for insertion orders) rather than just the .

5.4 *On-Line Credit Authorization Interface*



Our current product interface supports PCCharge Payment Server and ICVerify for real-time credit card processing from SunType.

These interfaces transaction processing software for the SunType Classified Advertising Systems™ is licensed to work with multiple users or a single workstation. This interface allows ad-takers to take credit card information and receive authorization for the charges while the customer is still on line.

5.6 *CallChecker (DoNot Call List)*



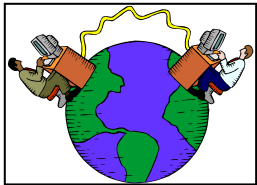
Suntype's new CallChecker™ program can be used to rapidly screen potential telemarketing calls for listing on up to 5 separate Do Not Call lists (i.e., national, state or company), as well as for existing business relationships from as many as 7 separate customer lists from systems within your company.

5.6 *Pagination*

The SunType ClassLink™ Xtensions provide a set of tools for using Quark XPress® 4-6 or Adobe Indesign® CS for desktop pagination and composition of display and classified line ads in conjunction with the SunType Classified System™. These Xtensions are available for either PC or MAC computers. The placement tools of the SunType ClassLink™ module allow users to flow and position classified ads generated by the SunType Classified Advertising System™, using the ClassLink output drivers. In-column classified ads, including both line (text) ads and in-line classified display, can flow into columns using Quark XPress® or Adobe InDesign®. Out-of-line classified display ads — typically multi-column true display ads — may be individually placed working from a palette of classified display ads in the ad run. The ad run can be generated using the SunType Classified System typesetting and/or proofing modules.

SunType's XTags text coding system decreases production time required for layout and pagination of classified sections by as much as half. The SunType Classified system's™ Xtag drivers allow the classified data to flow into QuarkXPress® or Adobe Indesign® layouts. Ads and classification headers retain all the text characteristics coded into them through the classified system, including — if desired — classified line ending and hyphenation decisions.

6. *HTML Support Package*



Converts classified ads to HTML. This package allows segmentation by classified, table of contents, links each page to the previous page, automatic inclusion inline classified display of ads and logos, places house and signature ads after number of ads-can be restricted to certain classes or ranges of classes. It uses report-generator-style variables to ad ID lines.

6.1 *Features*

➤ *Styling of Class Headers, Ad Breaks and Ad Texts*

Classification headers, ad breaks and ad texts are automatically assigned customizable HTML styles.

➤ *Optional Use of Netscape Extensions to Format Ad Text*

You may optionally specify use of the HTML extensions recognized by the popular Netscape™ web browser. If used, the Netscape extensions will result in ad texts that much more closely match their print counterparts. If the Netscape extensions are not employed, the driver will produce HTML code that will pass most HTML 2.0 or higher verification suites.

➤ *Segmentation By Class*

Ad runs are automatically broken into separate files by classification. Smaller file transfers place a lower burden on web servers and browsers.

➤ *Table of Contents Page Generation*

A table of contents page listing every classification with a hypertext link to that classes ads is automatically generated. This contents file will typically be a classified “home page”.

➤ *Placement of Next, Previous and Index Links at End of Each Class Page*

Each ad page has automatic hypertext links to the previous and next classification pages as well as the table of contents page. HTML Driver Documentation (6/14/00)

➤ *Character Set Translation*

Text is automatically converted to the ISO 8859 character set standard for HTML. Also, characters forbidden in HTML, such as angle brackets, are converted to their equivalent HTML entities.

➤ *Selective Replacement of Symbol Characters with Gifs*

HTML lacks any provision for symbol or dingbat fonts. The HTML driver helps solve this problem by providing for automatic substitution of text or images for selected characters in these fonts.

➤ *Automatic Inclusion of Classified Display Ads and Logos*

If you're using the classified system's display option, the HTML driver can automatically include inline classified display ads and logos. Just create a GIF file with the same filename and path as any print file specified in the display or logo record (but with a ".gif" file extension) and it will automatically be included in the run as an inline image.

➤ *Interleaving of House Ads or Copy*

Unlike a print page, the entirety of a web page isn't always visible. If a reader has scrolled part way down a longer web page of ads, the title and footer aren't visible. So, there may be nothing on the page to convey you're publication's identity, phone numbers and such.

To remedy this problem, the HTML driver can interleave a house ad or other copy every time a specified number of ads has been included. The interleave may also be restricted to certain classes or ranges of classes — like the personals.

6.2 *Program Components*

There are two program components, a new version of the ST_Type.DLL typesetting dynalink library and a new copy of the typesetting code format setup module, Setup.Exe. Place both in your classified system program directory and "HTML" will appear as output choices in both programs.

6.3 *Other Components*

- A sample configuration file, HTML.Dat, which should be placed in the classified data directory.

- Several decorative image files in GIF format, which may help get you, started in fashioning attractive classified ad web pages. Place these files, if you wish to use them, on your web server in a directory used for common image files.
- Several dingbat character images in GIF format to replace dingbat characters commonly used in print classified ad runs. These include star, phone and letter images. Place these files, if you wish to use them, on your web server in a directory used for common image files.

These files and their associated images are:

	d037.gif
	d038.gif
	d041.gif
	d043.gif
	d072.gif
	d074.gif
	d079.gif
	d080.gif
	d108.gif
	d111.gif
	d164.gif

Setup

7.1 *HTML.Dat File*

The appearance of your web pages may be customized by editing the contents of a text file, “HTML.DAT”, which should be located in the classified data directory. This file consists of several sections, each beginning with a colon and a keyword and followed by one or more lines of text.

You may edit HTML.Dat with any Windows or DOS text editor — just make sure that you save the file as plain ASCII text.

Note that most of these customizations require some knowledge of HTML.

A note on URLs (Uniform Resource Locators): Many of the customizations you might wish to do involve specifying URLs for text or image files. Remember to specify these URLs so that they’ll work on your Web server, which is nearly certainly different from DOS file naming conventions. Common mistakes include forgetting that the server operating system uses case-sensitive filenames or slashes rather than

You'll probably want to create a special directory on your Web server for commonly included image files. Then, the references to these image files can be consistent relative URLs.

7.2 *Class Header Style*

Specify the HTML style for classification header texts with the “:ClassStyle” keyword. For example, to specify “H2” for a class header style, place the line:

```
:ClassStyle H2
```

in HTML.Dat. The classad program will then place classification headers inside “<H2> ... </H2>” pairs.

If no ClassStyle is specified, H1 will be used.

7.3 *Classification and Ad Breaks*

Specify the breaks between ads with the “:AdBreak” keyword. For example, to set a horizontal rule 75% of the screen width, place the line:

```
:AdBreak  
<HR WIDTH=75% >
```

Note that the “WIDTH=75%” argument is a Netscape extension that may not work with some browsers.

If no AdBreak is specified, <HR> is used. The text of the ad break may be as many lines as necessary and ends with first line with a colon in column one.

Specify any code to start or a classification title with the keywords “:StartClassHd” and “:EndClassHd”. For example, to spice up the header appearance, you might use a fancy bar GIF:

```
:StartClassHd  
<IMG SRC="bar.gif" >  
<BR >  
:EndClassHd  
<IMG SRC="bar.gif" >  
<BR >
```

If these aren't specified, <HR> is used.

Ad breaks and classification headers start and end codes are placed in a <BODY> HTML context.

7.4 *Ad Pages Header and Footer*

The title of a web page is usually prominently displayed by web browsers. You may specify the title of ad pages with the keyword “:Title”. For example:

```
:Title Town Cryer Classified Ads
```

sets “Town Cryer Classified Ads” as the title for ad pages.

Many publications will wish to go much further in tailoring the appearance of their web pages. You might, for example, wish to place logos or messages at the top or bottom of ad pages.

You may specify the HTML code that starts and ends classified ad pages with the “:StartAds” and “:EndAds” keywords. After each one of these keywords, you may specify as many lines as you please. An example page header is:

```
:StartAds  
<!DOCTYPE HTML PUBLIC '-//W30//DTD W3 HTML  
3.0//EN' >  
<HTML> <HEAD> <TITLE>  
Town Cryer Classified Ads  
</TITLE> </HEAD> <BODY>  
<IMAGE SRC="cryer.gif" ALT="Town Cryer Classifieds">
```

An example page footer is:

```
:EndAds  
<H1> Call (000) 123-4567 to place an ad! </H1 >  
</BODY> </HTML >
```

If these aren't specified, innocuous HTML code is used as a default. Note that if you specify StartAds code, the “Title” keyword setting will be ignored.

7.5 *Previous, Index and Next Markers*

At the end of each ad page, the classified program places hypertext links to the next, previous and index pages. By default, these links are just text and you may wish to spice them up with icons or buttons. To do so, use the “:NEXT”, “:PREVIOUS” and “:INDEX” keywords to specify the images or text. An example is:

```
:Next <IMG SRC="docsrigh.gif" ALIGN=CENTER > Next  
:Previous <IMG SRC="docsleft.gif" ALIGN=CENTER > Previous  
:Index <IMG SRC="docscont.gif"  
ALIGN=CENTER > Table of Contents
```

If you use images only, remember to specify alternate text for text-only browsers. The specified link code and text will be placed in a “<A> ... ” context.

7.6 *Index Page Header and Footer*

As with ad pages, you’ll probably want to tailor the classified index page to project the identity of your publication. You may do so by editing the code that starts and ends the index pages. Use the keywords “:StartIdx” and “:EndIdx” on lines by themselves to specify that the following lines are code to start or end index pages. For example:

```
:StartIdx  
<!DOCTYPE HTML PUBLIC ‘-//W30//DTD W3 HTML  
2.0//EN’ >  
<HTML > <HEAD > <TITLE >  
Town Crier Classified Ads Table of Contents  
</TITLE > </HEAD > <BODY >  
<CENTER > <IMG SRC="crierlogo.gif" >
```

```
:EndIdx  
<P > <A HREF=" ../homepage.html" > The Crier Home  
Page </A >
```

7.7 *Class and Subclass Markers*

The classified program's HTML output driver creates the classification table of contents in the form of a compact definition list. Super and normal classes are listed in the term style (<DT>). Subclasses are listed in the definition style (<DL>). This list format was chosen because it's currently the only HTML list format that's fairly certain to be represented without bullets or other markers. That allows you to choose your own bullets or markers. To do so, use the ":SuperMark" and ":SubMark" keywords, following them with the text or code you wish to precede each classification's title. For example:

```
:SuperMark <IMG SRC="redball.gif" >  
:SubMark <IMG SRC="whitebal.gif" >
```

This puts a red ball bullet before super and normal class titles and a white ball bullet before subclass titles.

7.8 *Character/Image Substitution*

HTML currently does not include a specification for any dingbat, symbol or "pi" fonts. This poses a problem for publications that use a symbol font for decorative marks or to supply phone or letter icons — all of which are in common use in the publications likely to be posting classified ads on the net.

Our solution is to have the classified system's HTML drivers selectively substitute small inline images in GIF format for the missing characters. This works quite well so long as the GIF files are small and there are only a small number of characters being handled this way. If the symbol GIF files are large or you attempt to substitute images for a large number of characters, this solution will probably overload both web servers and browsers.

To apply character/image substitution, use the ":GFont" keyword once for each font for which you need to provide substitutions. Place the font name — which must have

the same name as used in the classified system font table — on the same line as the keyword and follow the keyword line with lines specifying character translations. For example:

```
:GFONT ZapfDingbats  
%= <IMG SRC="d037.gif" ALT="phone" >  
)= <IMG SRC="d041.gif" ALT="mail" >
```

specifies two character/image translations for the ZapfDingbats font. The (☛) d037.gif file will be used for the character in the “%” position, ☛, and the (☘) d041.gif file will provide the image for the character in the “)” position, ☘.. Note the use of the HTML “IMAGE ALT” argument to provide alternate text for text-only browsers.

7.9 *Interleaving House Ads*

You may specify HTML copy to be interleaved with ads with a command set like:

```
:INTERLEAVE count class-list  
The HTML content of the interleave,  
in as many lines as necessary.  
...
```

The count should be the number of ads between interleaves. It may range from 1 to 999.

The class list should be a list of classes and class ranges, separated by commas. For example:

```
:INTERLEAVE 4 1,100-110,2000
```

This calls for the following text to be interleaved every four ads in classes 1, 100 through 110 and 2000.

The interleave content may be any HTML that will be valid inside a <BODY> environment. You’ll typically want to end it with the same code you’re using for ad rules.

7.10 *Netscape Extensions*

The Netscape Communications Corporation has been advocating several extensions to HTML and have incorporated the ability to handle these extensions in their own Netscape browser. Many browsers don’t recognize the Netscape extensions, but most will ignore them.

The Netscape extensions include some very attractive options for formatting classified ads and, if you enable the classified system’s HTML driver’s use of the Net

scape extensions, your ad runs will have an appearance much closer to their printed counterparts. In particular, the output drivers will use the Netscape “” and “<CENTER>” to adjust ad text size and alignment. Turn on the Netscape extensions by placing the “:Netscape” keyword on a line of its own.

If you turn on the Netscape extensions, you should use the “:BaseSize” keyword to specify the size of the normal font used for classified body text. The HTML driver uses this figure for a baseline: for each two points of classified font size above or below the baseline, the font size level will be adjusted by one. The base size should be specified in decipoints (one-tenth points) on the same line as the keyword. For example:

```
:NETSCAPE  
:BaseSize 80
```

Tells the driver to use the Netscape extensions and that classified ad texts are normally set in eight-point type.

You may change the “every two points” behavior by specifying your own divisor in tenths of a point. For example, if adding an HTML size increment for every two points seems too extreme, you might add the line:

```
:Divisor 40
```

to specify an HTML size change for every four points (40 tenth points) of classified size change.

If Netscape extensions are turned on, the HTML output driver will also use HTML 3.0 table commands to produce boxed ads to match those produced by the PostScript and XPress drivers. The boxes will show up when Netscape 1.1 or later is used as a browser. Prior Netscape versions ignore the table commands with no negative consequences.

7.11 *Unique Ad IDs*

Every ad is ordinarily marked in the HTML output with a comment that looks like:

```
<!-- @ClassAd -->
```

If you're using the HTML driver to feed some sort of database, you may find it useful to add the line:

```
:UNIQUEIDS
```

to the HTML.DAT file. This will cause the ad comments to appear more like:

```
<!-- @ClassAd -->  
<!-- @ID 950612-00001.0001 0120 FOR SALENO -->
```

The first extra ID code is the ad's account code and item number. Together, they're unique to the ad. Following is the ad classification number (*four digits*) and sort key within the class (*ten characters, may include spaces*).

7.12 *File Name Conventions*

The conventional filename extension for HTML files is ".html". Some web servers require this extension for HTML files; others allow ".htm" or even ".ht".

Since DOS filenames allow only three character extensions, the HTML driver creates all its files with ".htm" filename extensions. However, it assumes that all these files are going to be renamed on the web server to have ".html" extensions. So, when it creates a hypertext link from one output file to another, it uses the ".html" filename extension.

If your web server will recognize ".htm" files as HTML, you may wish to change the way the output driver creates hypertext links. You may tell the driver to use ".htm" rather than ".html" in link codes by adding the directive:

```
:EXTENSION htm
```

to the HTML.Dat file. Note that there's no period in the directive's argument.

8 *Generating HTML Ad Runs*

Creating HTML ad runs should be simple. First, create a directory to receive the ad run, which may consist of several files. You won't want to pick them out, one-by-one, from the classified file directory. Then, run the classified system's typesetting or proofing modules as usual. When the classified program asks for choice of output table, drivers, etc., you should find an additional output driver option for HTML. Specify HTML output to file. For an output filename, specify the full pathname of the file you wish to be the table of contents page for the ad

run. You'll probably wish to use ".HTM" for a file extension. The ad run will be created in the same directory as the table of contents file. It will consist of multiple files named CLAS0001.HTM, CLAS0002.HTM, CLAS0003.HTM ... plus copies of any GIF files for classified display ads.

8.1 *WWW Posting*

The files created by the proofint or typesetting module should all be moved to the same directory on your web server. The hypertext links between the files are all relative URLs, so if they aren't all in the same directory, the links will be invalid. The ads run HTML files must also be placed so that any relative URLs pointing to standing items like image GIFs for characters or rules will be valid.

If the web server's file system accommodates file extensions greater than three characters, you must also rename all the transferred ".htm" files to have ".html" extensions. If you're using a server that has Perl available — as does most Unix servers — you may use the following Perl script to rename the files:

```
#!/usr/local/bin/perl
# Usage: htm2html [files]
if (!@ARGV) {
    die "Usage: htm2html [files]";
}
for (@ARGV) {
    $was = $_;
    s/\.htm$/\.html/;
}
```

You may need to edit the first line to correctly specify the path to the Perl binaries on your system.

Name this script "htm2html" and run it with a command line like:

```
htm2html *.htm
```

You may preface the "*.htm" with a file path.

9 *Sample*

Following is the contents of the HTML.Dat file used to produce the ad pages on the SunType WWW example:

((http://classifiedgroup.com/html/a1.html):

:NETSCAPE

:BaseSize 80

:ClassStyle H1

:startclassHd

:EndClassHd

:AdBreak <HR WIDTH=75% >

:GFont ZapfDingbats

% =

& =

) =

+ =

H =

J =

O =

P =

l =

o =

α =

:StartAds

<!DOCTYPE HTML PUBLIC "-//W30//DTD W3 HTML 3.0//EN" >

<HTML> <HEAD> <TITLE>

Suntype Classified HTML Output Test

</TITLE> </HEAD> <BODY>

:EndAds

</BODY> </HTML>

:StartIdx

<!DOCTYPE HTML PUBLIC "-//W30//DTD W3 HTML 3.0//EN" >

<HTML> <HEAD> <TITLE>

Suntype Classified HTML Output Test Index

</TITLE> </HEAD> <BODY>

<CENTER>

<p> <IM G SRC="suntype.gif" >

</p>

</CENTER> <H1> Classified Ads Index </H1 >

:EndIdx

<P> This classified ad run was produced 100% automatically - including the

ad index - in class-segmented, Web-ready HTML files
by the new HTML driver for the SunType Classified System, version 4.3.

```
</BODY> </HTML>  
:SuperMark <IMG SRC="redball.gif" >  
:SubMark <IMG SRC="whitebal.gif" >  
:Next <IMG SRC="docsrigh.gif" ALIGN= CENTER > Next  
:Previous <IMG SRC="docsleft.gif" ALIGN= CENTER > Previous  
:Index <IMG SRC="docscont.gif" ALIGN= CENTER > Index
```

```
:Interleave 2 1,100-110,1000  
<TABLE BORDER> <TR> <TD> <BR> <H3>
```

```
<CENTER> Messages like this may be <BR>  
automatically interleaved after <BR>  
every # ads. The interleave <BR>  
count is customizable.. </CENTER> </H3> </TABLE>  
<HR WIDTH=75%>
```

9.1 *Special Character Translations*

There are a number of characters that cannot be typeset from SunType using direct keyboard input because those characters are reserved for system functions. Prior to the advent of the Web, these keys had little typesetting impact (i.e., the tilde (~) character was rarely used as a stand-alone object prior to the Internet because it was normally combined with another character, as in "ñ").

SunType solves this problem by substituting other unused character values during ad entry and then translating them to the desired typeset character on output.

Some have been around for quite a while, and a few additional characters have been added lately to allow embedding of HREF codes into ads to create live links in ads sent to the Web.

Here is the list:

Character Desired	ASCII value	Win Char*
<	231	ç
\	221	Ÿ
^	226	â
{	222	¡
	223	■
}	230	µ

~	227	π
>	232	Φ
"	233	Θ

*These characters don't necessarily correspond to what you would get in SunType, which uses a different character complement when editing. Most will produce a Greek character of some kind.

You may choose to employ these characters as needed in SunType macros – this approach is necessary when working with HREF formatting strings. “ASCII” values are entered from the keyboard by holding the <ALT> key down and typing the number on the *numeric keypad*, not on the regular keyboard.

When doing HREF coding for the web, the macros for the print output are set up to use Font 98 {f98}, which is the non-printing (hidden) font in SunType. Examples of this setup are available from SunType support.

Index

A

Ad Breaks HTML 3-4
Ad Duplication Option 2-14
Ad Ids-HTML 3-9
Adobe Reader 1-3
ASCII Text 3-3

B

C

CallChecker 2- 19
Characters HTML 3-7
Characters translation 2-11
3-13
Class Header HTML 3-4
Command Line Flags 2-15
Counters Multiple 2-1
Counter Variable 2-1
Credit Cards 2-7

D

Dingbat Character 3-3

E

Email Number 2-4

F

Fax Number 2-4
File Name 3-9
Flag Ad Entry 2-13

H

Headline/Footer HTML 3-5
House Ads 3-7
HTML Ad Runs 3-10
HTML Features 3-1
HTML Components 3-2
HTML Index 3-5
HTML Output Driver 1-7
HTML Sample 3-11, 12
HTML Setup 3-3
HTML Support Package 3-1

I

Images HTML 3-7
Indexing HTML 3-5,6
Input User Defined Fields
2-14

Install Acrobat 1-3
Install V4.3 1-1
Internet Credit Card 2-7
Invoice Variables 2-6

L

Leading 2-11
Line Endings 1- 4, 2-3
Lose Columns 2-7

M

Markers 3-5,6
Multiple Counters 2-12

N

Netscape Extensions 3-8
New Features 2-14
No Detail Accounts 2-5
Numeric Sort 2-6

O

On-Line Credit 2-18

P

Page Header/Footer 3-6
Posting HTML 3-10
Pricing Line/Measure 2-1
Proofing Voice Box 2-17

R

Re-aging Utility 2-7
Report Generator 2-5

S

Seller Numbers 2-4
Special Character 2-11
Special Directory HTML3-3
Spell Check 2-18
STImport Menu Sample 2-10
STImport Utility 2-9
Styling HTML 3-1

U

URL's 3-3
User Defined Field 2- 3

V

Voice Box Proofing 2- 6

S

Sample HTML 3-11
Seller Numbers 2-4
Subclass Markers 3-6
System Specs 2-10

U

Unique Ad Ids 3-9
Using email and fax entry 2-4
User Fields 2-16

V

Variable Counting 2-12
Voice Box Proofing 2-6

W

www. Posting 3-10

Y

Year Numeric Sort 2-6

Z

Zones 2-3