

# Release Notes for Classified 4.2

---

## 1 Program Installation

---

This program set needs some files from the 4.1 set. So, install the 4.1 program set in a directory, then copy the files in this set over it. The program set is currently distributed in a zip archive spanning multiple floppy disks. To unpack and install the new program set, change DOS directory to the directory containing the 4.1 program set copy you wish to overwrite. Insert the last floppy disk of the set in a floppy drive and issue the command:

```
a:pkunzip a:42progs
```

The archive program will ask for the first disk in the set, then additional disks until program installation is complete.

### 1.1 Data Conversion

---

After backing up your classified file set, edit the file 41to42.bat to make sure the set command points to the data files you wish to convert. Then,

run 41to42.bat to convert files. Expect your data files to expand by 40-50% due to new indexes and expanded data structures.

The files VBoxes.Fil and CCardRec.Fil will disappear as they are absorbed into the classad and account files respectively.

## 2 New Ad Pricing Features

---

4.2 has new features that make it possible to price different lines or regions of an ad differently. These new features are available when the basic pricing mechanism is typeset line or measure.

### 2.1 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 “M COUNT1” will hold the measure of the first line; ”M COUNT2” will hold the measure of the body of the ad; and “M COUNT3” 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.

## 2.2 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 4.2 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.

For example, if an ad is coded:

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

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 body lines.

**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.

## 2.3 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 4.2 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.**

### 3.1 Requiring Input in User-Defined Field

---

4.2 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 mail boxes, 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 Miscellaneous New Options

---

### 4.1 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.

### 4.2 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 order then run “Reset Sort Keys” module to fix old sort keys.

### 4.3 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 date entered.

# SunType Classified System

## Version 4.2 Release Notes

---

---

Program Installation	1
Data Conversion	1
New Ad Pricing Features	2
Multiple Counters	2
Variable Counting	3
Change Flag in Ad Entry	4
Miscellaneous New 4.2 Features	5
Requiring Input in User-Defined Field	5
New Ad Duplication Option	6
Ad Duplication and Seller Numbers	6
Miscellaneous New Options	7
Spell Check	7
Year-Numeric Sort	7