

Keystroke™

Point Of Sale

Custom Modules and Advanced Features

Version 4

Copyright © by Specialized Business Solutions, 1988-1999. All rights reserved.

Keystroke is a trademark of Specialized Business Solutions.

All other products mentioned are trademarks or registered trademarks of their respective companies.

This User Guide, as well as the software described in it, is furnished under license and may be used or copied only in accordance with the terms of such license. The content of this User Guide is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by Specialized Business Solutions ("SBS"). SBS assumes no responsibility or liability for any errors or inaccuracies that may appear in this book. The copyrighted software that accompanies this User Guide is licensed for use only in strict accordance with the SBS License Agreement, which the Licensee should read carefully before commencing use of the software.



Specialized Business Solutions

<i>Service</i>	<i>R & D</i>
<i>PO Box 2019</i>	<i>2508 East Broadway</i>
<i>Dillon, Colorado 80435</i>	<i>Long Beach, California 90803</i>

Sales: 800•359•3458 970•262•1720
Software Maintenance Services: 800•275•4727
Basic Phone Support: 562•987•3459

Table of Contents

- Custom Modules** **1**
- GL-Link** **3**
 - Setup Menu** **4**
 - Link To Function
 - Parameters Function
 - Posting Accounts Function
 - Inventory Departments Function
 - Payment Types Function
 - Link Menu** **8**
 - Sales, AR Function
 - Using GL-Link with More Than One Store
 - Store Transfers Function
- Script Editor** **12**
 - Types of Scripts** **16**
 - Script Parameters** **20**
 - Script Menu** **24**
 - Clear Function
 - Edit Function
 - Delete Function
 - Recall Function
 - Save Function
 - Reports Menu** **25**
 - Special Menu** **26**
 - Copy Function – 
 - Paste Function – 
 - Pack Scripts Function
 - Export Function
 - Import Function
- Importer** **28**
 - Source Data Guidelines
 - Setup Menu** **29**
 - Import Menu** **31**
 - Troubleshooting Tips** **32**

Click on the Colored Text to Go to That Page

Production	33
Production Menu	36
Template Menu	36
Reports Menu	37
File Menu	38
Advanced Features Appendix	39
Protecting Your Investment	39
Network Version Features	44
Register Numbers	
Startup Switches	47
File Management	53
Subdirectories	
Temporary and Resource Files	
Data Files	
Managing Disk Space	
Printer Drivers	68
Customer Display Control Files	70
Field ID	
Display Strings	
Label Form File Structure	73
Header Parameters	
Field Parameters	
Sample Labels	
Other Form Files	77
Using Keystroke Data With Other Programs	78
ASCII Delimited Data Files	
Extracting Data From Keystroke	
Updating Keystroke Data Using Other Programs	
Direct Access from DOS Prompt	79
Pay Off AR Balances – ARPAYOFF.EXE	
Report Cues – RPT.EXE	
Export from DOS Prompt – KSEXPORT.EXE	
GL-Link from DOS Prompt – GLLINK.EXE	
Importer from DOS Prompt – IMP.EXE	
Summarize History – SUMMARY.EXE	
Auto-Production from DOS Prompt – PROD.EXE	
StoreLink Inventory Update – STRUPDTE.EXE	

Click on the Colored Text to Go to That Page

- Stand-Alone Utilities** **90**
 - Show Document – SHOWDOC.EXE
 - 4.0 Conversion – CONV40.EXE
 - Setup – SETUP.EXE
 - Program Scheduler – KSSCHDLR.EXE
 - Recurring Invoices/Recovery Merge – RECMRG.EXE
 - Print Labels from Invoice Form – FORM2LBL.EXE
 - Format Qty Column on Invoice Form – FORMQTY.EXE
 - Import Account Receivable Balances – IMPORTAR.EXE
 - Import Alternate Inventory Codes – IMPALTIC.EXE
 - Bike-a-Log On Disk Link – BODLINK.EXE
 - Delete Utility – DELUTIL.EXE
 - ChangeNums – CHNGNUMS.EXE
 - Print Unused Stock Numbers – NUMBERS.EXE
 - Check Inventory Script Flag – CHKSCRPT.EXE
 - Stamp – STAMP.EXE
 - Check Stamp – CHKSTAMP.EXE
 - Check Header Information on Transaction Files – HEADER.EXE
 - Remove Duplicate Sales Transactions – REMDUPES.EXE
 - Copy Next Number – COPYNUM.EXE
- Miscellaneous Files** **104**
 - Large Numbers – LRGPRINT.TXT
 - Screen Saver – SCRNSAVE.TXT
- Common Errors and Problem Solving Tips** **105**
- Error Messages** **106**
- Warning Messages** **109**

- Index** **111**

Click on the Colored Text to Go to That Page

Custom Modules

Custom modules are special add-on features for the **Keystroke POS** system which can be added to better accommodate the specialized needs of your business. If you wish to purchase one or more of these modules, please contact the dealer from whom you purchased the **Keystroke POS** program. The following modules are currently available for use with the **Keystroke POS** system:

GL-Link

The GL-Link module provides an interface to the general ledger module of third party accounting systems. Journal entries are generated for sales and accounts receivable transactions and can be output to a generic report format, directly into the data files of a specific GL software package, or to a file which is pre-formatted and ready to be imported into a GL package. GL-Link includes special data transfer options for: **BusinessWorks** by State of the Art, **PAS III** and **GBS II** by CYMA Systems, **QuickBooks** for DOS and Windows by Intuit, **AccPac** products by Computer Associates, and **Macola**. (Links for other popular accounting systems may also be available. Please contact your local Authorized **Keystroke** Dealer for more information.)

Script Editor

The Script Editor module provides the ability to set up customized entry procedures for inputting line items at the point of sale. These procedures, called "Scripts", are automatically run whenever scripted Inventory items are sold. Scripts can be used to accomplish a variety of advanced features such as: *reminder messages, prompting for the clerk to enter comments, components, automatic tag-along items, alternate items, superseded or replacement items, and/or suggested add-on items*. Each individual Inventory item may have its own unique Scripts, or the same Scripts can be used with numerous items.

Production

The Production module is a special Inventory maintenance tool designed to help manage items (quantities and costs) which are produced in batches from individually purchased components. An example of how the Production module can be used is to produce Gift Baskets, where you may purchase baskets and various components such as fruit and other edibles, and sell them as a finished good. The Production module can also be used to break down products from bulk units (as purchased) into smaller units to be sold. Production items and the appropriate quantity of each component item can be saved for repeated use.

Importer

The Importer module provides the ability to transfer data from an ASCII Comma (or Tab) Delimited file format into the **Keystroke** system. Inventory, Customer, and Vendor information may be imported as complete records (e.g., when converting from a different software program) or merged, in order to modify only specific fields in an existing database (e.g., price updates obtained on disk from a vendor).

The **Electronic Draft Capture (Keystroke EDC)** and **MultiStore** modules described below are each sold separately and are not included in the **Keystroke Custom Modules Bundle**. Separate documentation is shipped with each of these modules. Please contact your local Authorized **Keystroke Dealer** for more information on these add-on modules.

Keystroke EDC

The **Keystroke EDC** (Electronic Draft Capture) module adds the ability to perform electronic credit card or check authorizations and end of day funds processing. **EDC** may be used with either single or multiple register **Keystroke** systems. A magnetic stripe reader, modem, and third party processing service are required. **EDC** is compatible with nearly all major processors.

MultiStore

The MultiStore module includes a set of special functions designed to help manage sales and Inventory in businesses with multiple stores and/or warehouses. Primary features include: Quantity On Hand lookup for alternate stores/locations; inter-store Inventory transfers and composite Inventory level reports; and StoreMerge for transaction consolidation. Data is transferred between locations via a modem and third party communications software (e.g., Close-Up by Norton-Lambert), which must be purchased separately.

Installing Custom Modules

Custom Modules should *always* be installed at the same time you install the **Keystroke** program. A complete set of **Keystroke** program disks will usually consist of from 2 to 4 disks, depending on the number of Custom Modules you purchased. The program files for any Custom Modules will be included on either **Disk 2** or on a separate disk labeled "Custom Modules". Custom modules will automatically be installed along with the standard **Keystroke** program files.

***Note:** All **Keystroke** program files (including those for Custom Modules) should always be installed at the same time and from the same set of disks. Custom Modules will not run with the **Keystroke** program of a different version (i.e., files from Version 4.01 can not be intermixed with files from Version 4.05).*

Accessing Custom Modules

To access Custom Modules, first select Custom from the **Keystroke** Main Menu. If any Custom Modules are installed on the system, the names of those modules will be displayed on the screen. You can then select the module you wish to operate. If the Clerk logged in does not have sufficient Security Level to use Custom Modules, selecting Custom from the Main Menu will have no effect other than a warning beep.

GL-Link Module

The GL-Link module is used to consolidate **Keystroke** Sales and Accounts Receivable data into journal entries. These entries can either be printed out in a report format or saved to a file which can then be imported into the General Ledger of a separate accounting software program.

GL-Link's Standard Link creates an ASCII quote/comma delimited file of journal entries which can be imported into any General Ledger program that has a definable import function or most standard database, spreadsheet, and reporting programs. GL-Link can also create files specifically for use with the following programs: **BusinessWorks** Integrated Accounting by State of the Art, Inc. (800-447-5700), **PAS III** and **GBS II** by CYMA Systems, Inc. (800-292-2962), **AccPac** BPI/Bedford/Plus by Computer Associates International, Inc. (408-562-8800) the **Macola** Software, Inc. accounting package (800-468-0834), and **QuickBooks** (DOS and Windows) by Intuit, Inc. (800-446-8848).

The combination of **Keystroke**'s sales management and InventoryI control functions with the General Ledger, Accounts Payable, and Payroll modules of an accounting software package will provide a complete system with which to run your business. Because these are separate programs developed by different companies, the GL-Link module must be used to consolidate data between the two systems. GL-Link does not, however, transfer purchasing information to the General Ledger. Inventory account balances in the General Ledger must be adjusted using either the Accounts Payable module or by making appropriate journal entries for InventoryI received.

***Note:** GL-Link does not make adjusting journal entries to compensate for changes made to Invoices or ROA Payments that have already been imported and posted to the General Ledger. GL-Link also does not create journal entries for adjustments that are made directly to Inventory Database records (i.e., Quantity On Hand and Cost). Adjusting journal entries should be entered manually in order to compensate for such changes.*

BusinessWorks is a registered trademark of State of the Art, Inc. (800-447-5700).

CYMA PAS III and GBS II are a products of CYMA Systems, Inc. (800-292-2962).

ACCPAC BPI Accounting, Bedford, and Plus are registered trademarks of Computer Associates International, Inc. (408-562-8800).

QuickBooks is a registered trademark of Intuit, Inc. (800-446-8848)

GL-Link Menus

The GL-Link menu bar includes the Common options (see Introduction chapter) along with the menus and functions listed below.

Link Menu

Sales, AR send sales and AR entries to the GL

Transfers send store transfers (from StoreLink) to the GL

Setup Menu

Link To select type of interface (e.g., Report or specific GL software package)

Parameters enter Link settings such as Detail and data directories

Posting Accounts enter GL account numbers for journal entries

Inventory Departments enter GL account numbers for Departments

Payment Types enter GL account numbers for Payment Types

Setup Menu

Before GL-Link can operate, Posting Accounts and Parameters must be set up using their respective functions on the Setup menu. After all the parameters and account numbers are entered, you may link sales and accounts receivable records by selecting the Link – Sales/AR function from the Link menu. GL-Link will export the appropriate data for the selected date range to the link file specified in the Parameters function.

The link file will contain all necessary journal entries for the specified sales period. To post this information to the General Ledger, follow the appropriate instructions for importing journal entries to the General Ledger (described in the documentation of your accounting system). For example, if you are using BusinessWorks, you must then use the General Ledger – Transfer – Import Function to bring in the data.

Link To Function

Use the Link To function to select how you wish to link your data: to a specific accounting software program, by using the Standard Link, or as a Report.

- Report
- Standard Link
- Business Works
- CYMA IV
- CYMA PAS III
- CYMA GBS II
- AccPac BPI
- AccPac Bedford
- AccPac Plus
- Macola
- QuickBooks DOS
- QuickBooks Windows

If your accounting system is not listed, you may still be able to use the Standard Link or Report option. The Report option allows you to print out a listing of journal entries which can be manually input into your accounting system. The Standard Link places entries in a file which you may be able to import if your accounting system has the capability of defining the file format for its journal entry import function.

Note: The settings in the Parameters function will differ depending on which option you select in the Link To function.

Parameters Function

The Parameters function is used to enter the location where the journal entries created by GL-Link are to be stored and to change some of the options on how the entries are created. Options will vary depending on which accounting system you have selected in the Link To function. When requested in the dialog box, the following parameters must be entered in order for GL-Link to work properly:

Directory [] *The DOS location where the journal entry data file will be sent (saved)*

File Name [] *The name of the file in which to store the linked data (e.g., KEYSTROK.DAT)*

<Print To> *Screen, Printer, or File*

<Printer/File Name> *The name of the printer or file to which the report is to be sent*

<Detail> *Detailed, Inventory Summary, Summary by Day, or Totals*

Company Code [] *The 3-character code which identifies your Company
(CYMA accounting system only)*

Source Code [] *The 9-character Macola code which identifies the source of the GL data
(KEYSTROKE)*

Journal [] *The letter of the CYMA Journal to which entries will be posted (usually G)*

Journal Number [] *The number of the Journal in which entries will appear
(BusinessWorks only: we recommend the use of one of the
Special Journals, 10, 11, 12, or 13)*

[] **Suppress Zeros** *If marked ON, journal entries with a zero dollar amount will not be
created. It is recommended that you leave this ON to simplify reports*

[] **Update Lock Date** *if marked ON, the Transaction Lock Date/Time will be updated to the
Ending Date/Time of the period being linked*

The file link directory should be in the same location as the General Ledger data files for your company (e.g. C:\BW\SAMPLE). For AccPac BPI, the subdirectory under the BPIES\ directory is your company ID plus “.CC” (e.g. C:\BPIES\SAMPLE.CC).

The Detail setting allows you to select the level of detail that will be generated:

Detailed *creates a separate journal entry for each transaction*

Invoice Summary *summarizes the sales Invoices into a single entry for each day, but will
generate a separate entry for every type of transaction (layaways, ROA payments, etc.)*

Summary by Day *combines all the activity for each day into a single journal entry*

Totals *produces a single entry for the entire period (This will be a combination of all activity
over the specified period, and generates the minimum number of journal entries, which is
the same as the Grand Totals printed at the bottom when Link To is set to Report)*

After entering all the parameters, select [Ok] to save your entries. Once the entries have been saved, you will not need to use this function again unless you wish to change one of the entries.

Posting Accounts Function

Use the Posting Accounts function to enter the Account Numbers (from your Chart of Accounts) which will be used to create journal entries for **Keystroke** Sales and Accounts Receivable data. After selecting this function, a box will appear that lists the accounts for which you must enter account numbers. To enter an account number, move the cursor to the appropriate field and enter the number (up to 20 characters, alphanumeric). Only certain types of accounts should be used for each type of entry; the message line at the bottom of the **Keystroke** screen will indicate the type of account that may be used for each entry. Following are brief descriptions of each type of account.

Sales Subtotal used for Invoice Subtotals (before Discount, Shipping, or Sales Tax)

Discount used for amounts entered in the Discount field on Invoices

Shipping used for amounts entered in the Shipping field on Invoices

Sales Tax used for Sales Tax amounts on Invoices

Cash Payments used for Cash Payment Type amounts (on Sales Invoices and ROA Payments)

Credit Payments used for Credit Payment Type amounts

Other Payments used for Other Payment Type amounts

Inventory used for cost of Inventory On Hand

Cost of Goods Sold used to offset the Inventory account when items are sold

Finance Charges used for Finance Charges collected in Accounts Receivable
(Finance Charges that have not yet been paid are not included.)

Layaway Payments used for deposits made on Layaways (typically a liability account)

Order Deposits used for deposits made on Sales Orders

Paid Outs used for cash amounts entered in the Paid Out function
(may be left blank in order to not post Paid Outs)

Inventory Variance used to offset the Inventory account from posted Variances
(may be left blank in order to not post Variance transactions)

Note: Some of the above account types are used as defaults only and may not be used if corresponding account numbers are entered in the Inventory Departments and Payment Types functions (described later on). However, even if you are specifying alternate accounts for all Departments and Payment Types in the optional Inventory Departments and Payment Types functions, it is still recommended to enter valid account numbers for all possible entries in the above fields. Exceptions to this rule are if your business does not have any use for certain types of entries (e.g., if you do not maintain an Inventory, the Inventory and Cost of Goods Sold may be left blank). Similarly, the Paid Outs and Inventory Variance fields may be left blank to intentionally not post such entries.

It is extremely important to enter the account numbers accurately and in the exact format that is required by your General Ledger program. **Keystroke** cannot validate the account numbers entered; therefore, incorrect account numbers will either show up as an account number error or as an unbalanced journal entry in your General Ledger program. The best way to enter your account numbers is to use a printed copy of your Chart of Accounts to look up the correct format for each account number. If you are not linking directly into another accounting package (e.g., printing a report, or using the Standard Link for use with a spreadsheet program), you can enter descriptive terms instead of account numbers.

If you are using either BusinessWorks or CYMA (after settings are correctly entered in the Setup - Parameters function), you may press **[INS]** to select account numbers from a list box containing the current Chart of Accounts from your general ledger.

After entering all the necessary account numbers, select the **[Ok]** button to save your entries. Once the entries have been saved, you will not need to use this function again unless you need to change account numbers.

Inventory Departments Function

The Inventory Departments function is used to define alternate Account Numbers for the sales of items from different Departments. Items sold that do not belong in one of the Departments entered in this table will use the default account numbers (as entered in the Posting Accounts function). You can post several different Departments to the same account number by putting each Department on the list and entering the same account numbers for each Department. This allows you to have more detailed Departments in **Keystroke** while having only a few major Departments on your financial statements.

Department *the Inventory Department from which the sales information will be posted*

Sales *account number to which sales information will be posted
(reflects the actual price of the items sold)*

Inventory *account number to which current asset information will be posted
(reflects the cost of the items sold)*

COGS *account number to which Cost of Goods Sold information will be posted*

For each Department entered on this list, a valid account number must be entered in the Sales column; however, for Non-Stock items such as Labor or Service (which do not have an intrinsic value) you may leave the COGS and/or the Inventory columns blank. Press **[INS]** to look up Department Codes or to select Account Numbers from the Chart of Accounts.

For sales, the Department used will be the Department listed on the item at the time the item was sold. Changing the Department on an Inventory item will not affect the posting of sales already entered; however, when posting a Variance, the program will use the Department currently entered on the Inventory item.

Payment Types Function

The Setup – Payment Types function can be used to specify separate Posting Account Numbers for each individual Sales Payment Type. If a Payment Type is entered with a valid Account number in the Payment Types function, the program will use that Account number. If a Payment Type is not included on this list, the program will use the appropriate default Account Number (i.e., for Cash, Credit or Other Payments) as entered in the Posting Accounts function. This is similar to Inventory Departments.

Number/Payment Name *the Sales Payment Type for which information will be posted*

Account *the account number to which payment information will be posted*

With the cursor in the Number column, press **INS** to select the desired Payment Type from the list box. Enter the appropriate Posting Account Number in the Account column. Repeat as necessary for each Payment Type. Press **F10** to save changes and exit.

Link Menu

The Link Menu accesses the final stage of transferring information from **Keystroke** into the General Ledger of a third-party accounting package: linking Sales and A/R data for a specified Date/Time range.

WARNING: *GL-Link does not check for the transfer of duplicate data; therefore, the Date/Time range should never overlap with that of a preceding link. GL-Link does not transfer Purchase Orders, nor does it make adjusting journal entries for changes made directly on Inventory Database records or for changes made to transactions that have been previously linked to the General Ledger. Such entries should be made either through the Accounts Payable module of your accounting system or by making manual journal entries in the General Ledger.*

Sales, AR Function

The Sales/AR function creates a special file which will contain journal entries for Invoices and ROA Payments dated within the specified Date/Time range. This file will be formatted and named according to the information entered in the Setup – Parameters function. After exporting data using the Sales/AR function, journal entries may be posted to the General Ledger using the Import function in your accounting program.

Linking with Specific Programs

Each General Ledger program has a unique way of posting the data file containing the new journal entries. Below are specific instructions on how to import the file from some of the packages supported by **Keystroke**:

WARNING: Since most accounting systems do not let you easily undo an Import or Posting function, it is important to make a backup of your accounting data **BEFORE** posting the journal entries created by GL-Link. We recommend you make a backup first, import/post the data from **Keystroke**, and then immediately check to make sure the posted entries are correct. This way, if something goes wrong, you can easily restore the backup of your accounting data without losing valuable information.

Link to Report

Depending on the parameters settings, the report will generate a list of the journal entries either on your screen, sent to a printer, or sent to a file. This listing can be very helpful when identifying what is going to be posted to your General Ledger, or can simply be output as a report to give to your accountant/bookkeeper for manual entry into another program.

Link to Standard Link

The Standard Link sends the journal entries to the ASCII Delimited file specified in the Parameters function. Each record in this file will contain a single entry consisting of the following fields: Sequence Number (used to group the entries together), Date (date of link), Transaction Type (e.g., Invoices, A/R Payments, Variance, etc.), Reference (e.g., Customer Number, Invoice Numbers), Account (as defined in the Account Numbers function), and Amount (dollar amount to be posted).

Link to BusinessWorks

To transfer the entries created by GL-Link into BusinessWorks, use the Import function located in the General Ledger of BusinessWorks. To do this, get into the General Ledger of BusinessWorks and select the Import Journal Entries function on the Transfer menu. Specify the Path and File name to be imported (these will be the same as the Directory and File Name that you entered in the GL-Link – Setup – Parameters function in **Keystroke**).

Link to CYMA PAS III and GBS II

Since CYMA General Ledgers do not possess an Import function, the journal entries are posted directly to the General Ledger Recurring Entries data file. To post these entries to the journal, select the following menu options:

- C. Record Journal Entries
- B. Add, Change or Delete Recurring Entries to review the entries
- OR
- C. Generate Recurring Entries into Journal to post the entries to your journal

Link to AccPac BPI

If you are using AccPac BPI Accounting, you must first set up the BPI program to accept the data from **Keystroke**. To do this, type “**PREPBPI**” at the “**KEYSTROK**” prompt (you only have to do this once). This batch program will copy the files IMPORTGA.TXT and GJPOST.KS to the BPI program directory “\BPIES”.

Note that BPI does not automatically place journal entries in the appropriate accounting periods (i.e., does not check the date of the entry). Therefore, in order for transaction journal entries to be placed in the correct periods, it is necessary to link data for one period at a time (the current period in BPI). Before importing journal entries into BPI, be sure to specify the appropriate period using the Select Company function in BPI.

Link to Macola

Before importing journal entries into Macola, the current transaction file should be posted in order to clear it. You can then proceed with posting the journal entries.

Link to QuickBooks® for DOS/Windows

The GL-Link interface with QuickBooks® is similar to that for other general accounting packages; however, due to certain restrictions in the QuickBooks program, several special considerations must be addressed to enable journal entries to be properly imported. First, you must set up an account in QuickBooks named “Keystroke Sales” with the type set to “Checking”. This account is used to record entries from GL-Link, and should always have a zero balance.

QuickBooks does not allow data to be transferred into any account with an account type of A/R (Receivables) or A/P (Payables). You must create a new account for Accounts Receivable and set its type to “Current Asset” instead of A/R. Similarly, you must create a new account for Sales Tax and set its type to “Current Liability” instead of A/P. Since **Keystroke** does not allow entry of account numbers greater than 20 characters, you must limit your Account Names in QuickBooks to no more than 20 characters.

The only peculiarity which can be confusing when setting up GL-Link for QuickBooks is that QuickBooks uses the term “Account Name” for what most accounting systems refer to as “Account Number”. These are essentially the same thing (i.e., enter the appropriate QuickBooks Account Names in the Account Number fields in GL-Link).

Importing GL-Link data into QuickBooks®

After running the Link – Sales, AR function in the GL-Link module, the resulting data may then be imported into QuickBooks. In QuickBooks, select the following menu options:

7. Set Up/Customize
2. Back Up/Restore/Copy Company Files
6. Import Company Data

Enter the name of the file created by GL-Link (GL-Link Setup menu – Parameters – File Name). Specify “No” to Special handling for transfers. Journal entries created by the GL-Link module will then be imported into your QuickBooks system.

Using GL-Link with More Than One Store

GL-Link can be used to combine individual sales information from different stores into your General Ledger and produce a consolidated financial statement for your entire organization. Since GL-Link creates a data file of journal entries which is then imported into your General Ledger, it is possible to create the file on one machine and then transfer (via modem or diskette) to the master machine (the one containing your General Ledger system) for importing. Each store will use its own parameters (file name, account numbers, etc.), thus allowing the posting of each store's information into different account numbers (usually called department by the General Ledger program).

Transfers Function

If you are using the **Keystroke** MultiStore module, the Store Transfers created by the StoreLink part of this module can also be posted to your General Ledger. This is done by using the Transfers function on the Link menu. This function will only appear if the Current Store is Number 0 (zero) and StoreLink files are available, and will transfer the Inventory Amounts from one store's account number to another's. Account Numbers for To and From will be read from the Inventory Accounts in the GL-Link setup data file (GLLINK.DAT), located in the data subdirectory for each store.

Script Editor Module

The Custom module **Script Editor** is used to set up customized entry procedures for inputting specific line items on sales transactions. Such customized line item entry procedures (Scripts) may include any combination of reminder messages, on-screen prompts, tag-along items, components, alternate items, and/or suggested add-on items (all of which are unique to each individual Inventory record).

Introduction to Scripting

Scripting is a powerful feature of the **Keystroke** system that encompasses a number of special uses. Some common terms used to describe what Scripting can do include: *kitting*, *parts assembly*, *sub-assembly*, *suggested items*, *tag-along or add-on items*, *superseded or replacement items*, *substitutes* and *alternate items*. Below is a sample of a Script in action, followed by a few other examples of how you might be able to utilize **Keystroke's** powerful Scripting features.

Class	Description	Qty	Price	Amount	T
ACCS-B110	Did they purchase paper clips?			5.99	T
ACCS-B110	52 PAPER CLIPS - 100 CT		0.74	2.36	T
PRINL32				510.44	I
COMP-110				0.00	T
ACCS-C160				3.49	T
SubTotal				522.28	
Misc.				0.00	
Tax / .5%				59.17	
Total				561.45	

Keystroke's Scripting feature provides the ability to streamline point of sale procedures for many types of specialized applications.

Keystroke Script in action

Scripting is ideal for:

- Simple Reminder Messages for individual items that need special attention when sold (e.g., special order items)
- Special instructions to prompt the Clerk to enter a Comment on line items (e.g., "Enter model of the item and description of the problem to be repaired")
- Tag-along items which are automatically added to the sales Invoice for items that are always accompanied by one or more supplemental items (e.g., installation, or core charges)
- Assembled or "kit" items that include sub-items or components (e.g., a gift basket which includes a number of other stock items). The items that make up the kit can either be added automatically, or the Clerk can be prompted to select each part from a list of available choices. The individual items that make up the main item may either be included or omitted from the printed Invoice

- Reminding the Clerk to suggest additional items to the customer, based on items that are commonly sold together (e.g., when selling a computer printer, a Script might suggest a cable and additional ribbons or toner)
- Superseded or replacement items that are virtually identical products, but labeled with different UPC numbers (e.g., liquor bottles in holiday packaging)
- When an item is out of stock, Scripts can be used to suggest one or more alternate or substitute items

Using the Script Editor module, you can create one or more unique Scripts for each individual Inventory item (Stock Number) in your database. Once a Script has been set up for an item, that item is referred to as a *Scripted Item*. Whenever a Scripted Item is entered on a sales transaction, the associated Script procedures are automatically initiated.

Scripts may be different for each item, or they can be copied from one item to another. Each Scripted Item may include up to 40 consecutive Scripts, which may consist of any combination of Reminder Messages, Tag-along Items, etc. Scripted items may also launch additional Scripts that have been set up for other items, (i.e., a Tag-along Item may have its own Script). For example, when selling a computer, the Script can suggest the sale of a printer, which in turn may be Scripted to suggest the sale of a printer cable, and provide the Clerk with a selection of items to choose from.

Setting Up Scripted Items

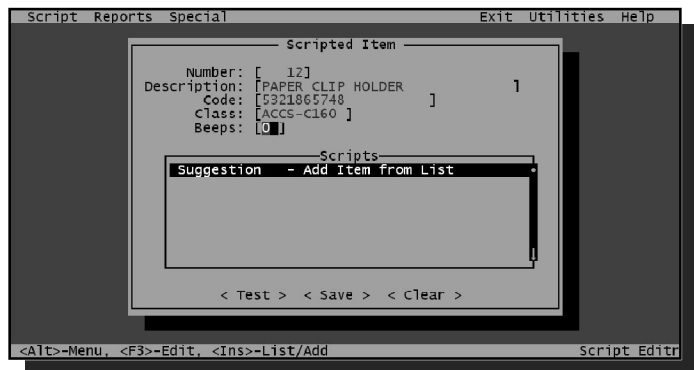
From the Main Menu, select Custom – Script Editor to access the Script Editor module. The Scripted Item dialog box will be displayed, from which you will be prompted to specify the item to be Scripted.

Note: *The initial cursor location is determined by the default item entry field as defined in the Parameters – Entry function in the Sales Manager. Once you have entered the appropriate information in one field, **Keystroke** will fill in the rest of the Database information for you.*

Enter the number of Beeps which notify the user when a Scripted Item has been entered.

Enter up to 40 Scripts per item.

Scripts can be tested from the set up screen.



Scripted Item dialog box

Number

Enter the Stock Number of the item for which you wish to set up the Script. You can either enter the appropriate number or press **[INS]** to select the item from a list box. If any Scripts have been previously set up for the Scripted Item, they will be displayed in the Scripts list box.

Description

Enter the Description of the item for which you wish to set up the Script. You can either enter the appropriate number or press **INS** to select the item from a list box. If any Scripts have been previously set up for the Scripted Item, they will be displayed in the Scripts list box.

Code

Enter the Product Code of the item for which you wish to set up the Script. You can either enter the appropriate number or press **INS** to select the item from a list box. If any Scripts have been previously set up for the Scripted Item, they will be displayed in the Scripts list box.

Class

Enter the Class of the item for which you wish to set up the Script. You can either enter the appropriate number or press **INS** to select the item from a list box. If any Scripts have been previously set up for the Scripted Item, they will be displayed in the Scripts list box.

Beeps

After specifying an item to be Scripted, enter the number of Beeps that will sound when the Scripted Item is entered on a sales transaction. Enter “0” (zero) for no sound, or up to 99 for a really obnoxious set of beeps. In most applications, 1 to 5 is plenty.

Adding Scripts

To add a new Script, position the cursor in the Scripts list box and press **INS**. This will display the Edit Script dialog box, enabling you to begin setting up a Script. Enter a Name for your Script and save it as a Reminder Message by pressing **F10**. This will take you back to the Scripted Item dialog box.

The maximum number of Scripts per item is 40 (although you’ll probably never need more than a few for a single item). When a Scripted Item is entered on a sales transaction, each Script is executed consecutively in the order in which it appears on the Scripts list. To add additional Scripts, position the highlight bar in the position where you want the new Script to be added and press **INS**; the new Script will be inserted above the current cursor position. Scripts can also be repositioned using **SHIFT DEL** to “cut”, and **SHIFT INS** to “paste” the Script above the current cursor position. To erase an individual Script, highlight it and press the **DEL** key.

Copying Scripts

If you have multiple items for which the Scripts will be identical or very similar, you can save time by using the Copy and Paste functions from the Special menu. Use the Copy function to specify which Scripts to copy, and the Paste function to append those Scripts onto a different item.

To copy Scripts from the current Scripted Item into the computer’s memory, press **SHIFT F8**. Next, bring up the item to which you want to copy those Scripts and press **F8**. For more information, see the full description of the Copy and Paste functions later on in this Section.

Editing Scripts

To edit Scripts that have been previously entered, move the highlight bar to the Script you wish to edit and press either **F3** or **ENTER**. This will bring up the Edit Script dialog box in which you will see the current settings for the selected Script.

Testing Scripts

Any time you add or edit Scripts for a Scripted Item, it's a good idea to test them out to make sure everything works the way you intended. To test Scripts, specify the Scripted Item and select the **< Test >** button from the bottom of the Scripted Item dialog box. This will launch the Scripts the same as if you were entering the Scripted Item on a sales transaction.

If all goes well, a "**Script Test completed**" message box will be displayed at the end of the test. The only portion of the Script the Test function does not check is how items are entered on the transaction, (i.e., add-on, tag-along, component, alternate, or replacement items). To test how such items are being handled, you will need to go to the Sales Manager and actually enter the Scripted Item on a new sales transaction.

Note: Do not set up Scripts that result in infinite loops or ridiculously long Scripts, (e.g., successive Tag-along Items that are themselves Scripted Items). If you accidentally set up an infinite loop Script, pressing **CTRL+END** will cause the system to immediately drop out of the Script.

Saving Scripted Items

To save new or edited Scripts, either press **F10**, select the **< Save >** button at the bottom of the Scripted Item dialog box, or select the Save function from the Script menu.

Setting Up Scripts

How each Script will operate is defined by several settings that are specified on the Edit Script dialog box as shown below. This box is displayed whenever you choose to add or edit a Script from the Scripted Item dialog box.

Each of the Script Types can be used in many different ways depending on your needs.


The parameters available for defining how each Script will function are slightly different for each Type of Script.



Edit Script dialog box

The **< Name >** setting provides you with a way of keeping track of how each Script operates when you are editing Scripted Items. **Keystroke** does not use the Script Name for any other purpose. You are not required to enter a Name; however, it may be helpful in locating a Script if you later need to review or edit it

The < **Type** > setting determines what the Script will do, and especially affects which parameters will be available to specify exactly how the Script will work. To specify the Script Type, select the < **Type** > button and select one of the Script Types from the list.

To save a new or edited Script, press  or select the < **Ok** > button at the bottom of the Edit Script dialog box.

Types of Scripts

There are five Types of Scripts from which to choose:

Reminder Message *used to prompt the Clerk with a text message*

Prompt for Comment *allows you to prompt the Clerk to enter a Comment on the Scripted line item and/or to initialize the comment to contain certain text*

Tag-along Item *allows you to specify a single item that will automatically supplement or replace the Scripted Item*

Add Item from List *used to give the Clerk a choice between several items that will either supplement or replace the Scripted Item**

Add Item from Range *used to give the Clerk a choice between several items that will either supplement or replace the Scripted Item**

** The main difference between these two is in how the optional items are defined (List vs. Range).*

Each Type of Script is defined by a number of different parameters; the set of parameters available for each Script Type will vary depending on the Script's purpose. The upcoming pages of this Appendix will discuss each Type of Script, including descriptions of each of their respective parameters.

Reminder Message

A Reminder Message is a simple on-screen text message used to communicate information to the Clerk. A Reminder Message Script might commonly precede a Tag-along or Add Item Script to provide special instructions to the Clerk.

To set up a Reminder Message, select the < **Message** > button and enter your desired text message. If the Reminder Message is to be used only when the Scripted Item is out of stock, turn ON the Out of Stock Only parameter by placing an “X” in the preceding brackets.

Note: *Reminder Messages and Prompt for Comment Messages will be displayed using the color setting for “Screen Flags”. You can change this setting using the Colors function on the Setup menu in Configuration Manager.*

Prompt for Comment

A Prompt for Comment Script is used to prompt the Clerk to enter a line item Comment for the Scripted Item. This Script Type is identical to the Reminder Message except that in addition to a text message, the line item Comment box is automatically displayed so the Clerk can be instructed to enter information (such as delivery or service instructions).

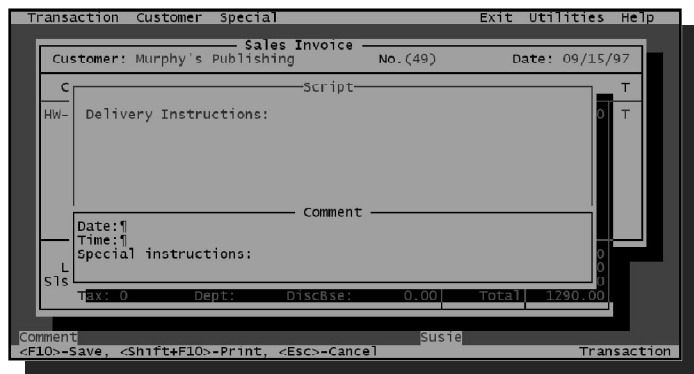
To set up a Prompt for Comment Script, select the < **Message** > button and enter your desired text message or special instructions for the Clerk, such as indicating what additional information should be entered in the Comment field.

If you want the line item comment to already contain information, select the < **Comment** > button. Any text entered in the Comment option will be added to the line item Comment on the transaction. The < **Comment** > button is available only on Prompt for Comment type Scripts; any information entered here will be added to the Line Item Comment when the Comment is displayed. This can be used to pre-enter repetitive information, or to create a form which the Clerk can fill in.

One special situation for this type of Script: if you leave the Message blank on the script, the program will display the Comment information, but won't give the clerk a chance to review or edit it; it will simply be added to the line item. This is very similar to adding a Comment to the Inventory item itself and then turning on the Copy Inventory Comments parameter in the Sales Manager. The advantage of using the Script is that the Copy Inventory Comments parameter affects all Inventory items, while Scripts allow you to select which items get an automatic Line Item Comment. This leaves the Inventory Comment available for other purposes.

If the Prompt for Comment is to be used only when the Scripted Item is out of stock, turn ON the Out of Stock Only parameter by placing an "x" in the preceding brackets.

A Prompt for Comment Script prompts the Clerk to enter additional information such as item serial numbers or delivery instructions.



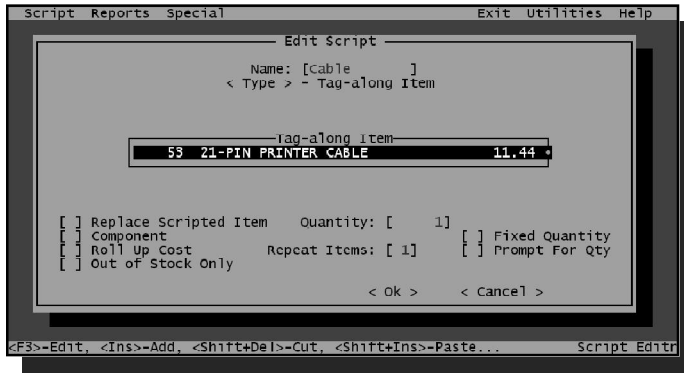
Prompt for Comment Script

Tag-along Item

A Tag-along Item Script is used to specify a single item that will automatically supplement or replace the Scripted Item. This Type of Script is typically used for kitting or sub-assembly items, where the Scripted Item always consists of components or additional items. The Tag-along Item Script can also be used to automatically handle replacement or superseded items, such as a bottle of liquor which may have two different UPC codes due to holiday packaging. (**Keystroke**'s Alternate Inventory Codes function can also be used for this same purpose).

The main element of a Tag-along Script is the item itself, which is the supplemental item, component, or replacement for the Scripted Item. To specify the tag-along item, move the cursor to the Tag-along Item field, and press **[INS]** to bring up a list box of your InventoryI records. Then simply select the appropriate item from the list box. Press **[TAB]** to move out of the Tag-along Item field.

If the Tag-along Item is to be used only when the Scripted Item is out of stock, turn ON the Out of Stock Only parameter by placing an "x" in the preceding brackets.



Tag-along Item Script parameters

Each of the Script Types can be used in many different ways depending on your needs.

The parameters that determine how a Script will function are different for each Type of Script.

Exactly how a Tag-along Item Script will work is controlled by the parameters shown above. Each of these parameters is described later on in this section under the Script Parameters heading.

Add Item from List (or Range)

The Add Item from List and Add Item from Range Script Types are used to give the Clerk a choice between one or more items that will either supplement or replace the Scripted Item. These Types of Scripts are basically the same as Tag-along Item Scripts, except they provide an opportunity for the Clerk to specify the supplemental, replacement, or component item(s). These Script Types would also typically be used for kitting or sub-assembly items, as well as for alternate items.

The only difference between these two Script Types is in how you define the set of optional items the Clerk may select from. If there is no obvious relation between the items in terms of Stock Number, Description, Product Code or Class, you'll need to specify the items one at a time using the Add Item from List Type (as shown below).



Editing an Add Item from List Script

Enter up to 60 individual items to be included in the list of items from which the Clerk can select.

The Min/Max Items parameters can be used to limit how many items may be selected. These particular settings will require that 5 items be selected.

Otherwise, if the desired items all start with the same characters in the Description, Product Code, Class or Manufacturer field (or if they are a consecutive range of Stock Numbers), the Add Item from Range Type may be more appropriate.

An advantage to the Range Type is that new Inventory items that fit into the specified Range are automatically available to the Script. With the List Type, individual items may be listed in any order you wish; however, the item list is limited to 60 items, and items may only be added to or deleted from the list by manually editing the Script.

Item List

When using the Add Item from List Script Type you can specify up to 60 individual items to be included in the item list. Items will be displayed in the order you enter them in the area titled “**Individual Items**”. These items are not automatically sorted, so when you add items to the list, be sure to position them in the desired sort order. The actual items that make up the item list are entered in the Individual Items list box, and the Display parameter allows you to specify how the items will be shown in the item list.

To set the primary Display field, select the < **Display** > button and choose Description, Product Code, Stock Number, Class or Manufacturer. The field you choose will be displayed to the left of the item list. Additional data in the item list will vary with each of the five choices.

To enter individual items, move the cursor to the Individual Items area and press **INS**. This will display a list box of your entire InventoryI, from which you can select an item. The selected item will be inserted on the item list above the cursor position. Continue adding all necessary items up to a maximum of 60 per Script. Items can also be repositioned on the Script using **SHIFT+DEL** to “cut”, and **SHIFT+INS** to “paste” above the cursor position. To erase an individual item, highlight it and press the **DEL** key. To move out of the item list, press the **TAB** key.

Item Range

The Add Item from Range Script Type allows you to specify a virtually unlimited number of items for the item list. Also, with this Script Type, the list from which items are selected is a full service list box, (i.e., automatic sorting and Quick Search). The < **Sort By** > parameter allows you to specify the default sort category for the item list.

The Sort By category determines how the Range is to be entered and how the item list will be sorted.

*Use the < **Test Range** > button to preview the item list you ve defined.*




The screenshot shows a window titled 'Edit Script' with a menu bar (Script, Reports, Special, Exit, Utilities, Help). Inside, there's a 'Replacement Item' table with columns: Description, Product Code, QOH, and Base Price. The table contains five rows of Acer laptop specifications. Below the table is a 'Quick search' field and 'Ok'/'Cancel' buttons. A footer bar shows keyboard shortcuts: <Enter>-Select, <Esc>-Exit, <Ins>-Add, <F3>-Edit, <F1>-Help, and 'Script Editor'.

Description	Product Code	QOH	Base Price
ACER P100-8MB-1GB-4XCD-SVGA	2254856987	1	\$1399.00
ACER P133-16MB-2GB-6XCD-MVGA	5157485632	6	\$1699.00
ACER P166-16MB-2GB-8XCD-EDOV3D	2245854856	2	\$2099.00
ACER P166-32MB-3.2GB-8XCD-EDOV	2387625147	6	\$2299.00
ACER P200PRO-32MB-3.2GB-8XCD	2541586963	9	\$2699.00

Editing an Add Item from Range Script


To set the default sort category, select the < **Sort By** > button and choose either Description, Product Code, Stock Number, Class or Manufacturer. The Sort By category also determines how the item Range is to be entered. For example, if Sort By is set to Class, enter the Class of the first item to be included on the item list in the Range Start field; enter the Class of the last item to be included in the Range End field. To check your Range settings, select the < **Test Range** > button. This will display the item list (with the Message in the top border) exactly as it will appear when the Script is executed.

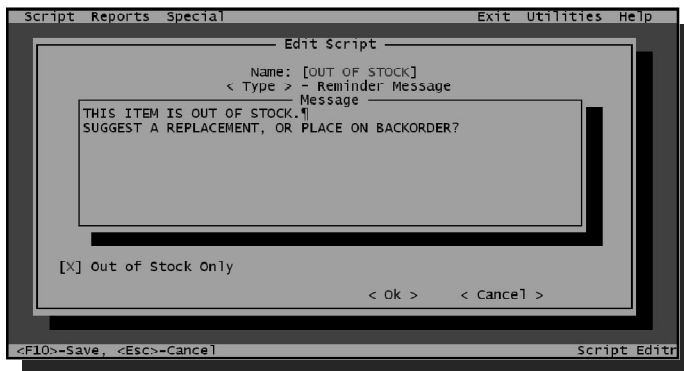
Script Parameters

The following parameter settings are used to define exactly how each Script will work. The parameters that are available for each Type of Script will vary, (i.e., not all parameters are available for all Script Types). The Replace Scripted Item, Component, Roll Up Cost, Out of Stock Only, Fixed Quantity, and Prompt For Quantity parameters are “toggle” fields, which may be turned ON or OFF using the  key.

Message

The < **Message** > button is used to enter a basic text message (e.g., special instructions or list box title) on all Script Types except Tag-along Items. Because Tag-along Item Scripts are automatic (require no intervention by the Clerk), there is no opportunity for a Message to be displayed.

To set up a Message, select the < **Message** > button and enter your desired text message. Press  to save the text Message. Below is an example of how the Message for a Reminder Message Script might appear on a sales transaction.



This Reminder Message Script would be launched only when the Scripted Item is out of stock.

Another Script might follow that lists comparable items that can be substituted.

Example Reminder Message Script

The message on Reminder Message and Prompt for Comment Scripts may be *up to 8 lines* long. The message on Add Item from List and Add Item from Range Scripts is actually displayed *in-line* with the top border of the item list box; therefore, the Message on these Script Types is limited in length. The Message on Add Item from List Scripts may be *up to 40 characters* in length; the Message on Add Item from Range Scripts is limited to *20 characters*.

Note: *You may enter longer text Messages; however, only the number of lines or characters as described above will be displayed. If you require a longer, more detailed Message than allowed for an individual Script, try using multiple Reminder Message Scripts. You can stack up to 40 Scripts per Scripted Item.*

Comment

The < **Comment** > button is available only on Prompt for Comment type scripts. Any information entered here will be added to the line item Comment when the Comment is displayed. This can be used to pre-enter repetitive information or to create a form which the Clerk can fill in. If you want the line item Comment to already contain information, select the < **Comment** > button. Any text entered in the Comment option will be added to the line item Comment on the transaction.

Replace Scripted Item

The Replace Scripted Item parameter will cause the Scripted Item to not be recorded on the sales transaction, and will instead be replaced by the item in the Script (either a Tag-along Item or the item selected for an Add Item from List/Range Script). This would typically be used to offer (or force) an alternate item to be entered on the sales transaction (e.g., a bottle of liquor which has two different UPC codes due to holiday packaging, or to suggest an alternate item when an item is out of stock). The Replace Scripted Item parameter is available on the following Script Types: Tag-along Item, Add Item from List and Add Item from Range.

Depending on the parameter settings for this Script, the item selected from this box might be an add-on item, a replacement, or a component.

The screenshot shows a window titled "Edit Script" with a menu bar (Script, Reports, Special, Exit, Utilities, Help). The main area contains the following text:

Name: [PICK]
< Type > - Add Item from List
< Message > - PICK 5 ITEMS FROM THE LIST
< Display > - Description
Individual Items

BATH BEADS	09327409349	1.28
BODY OIL	23059095089	1.96
FRUIT CANDY	2309570973	0.64
HERSHEY CHOCOLATE w/ ALMONDS	9780553347012	0.55

Below the table are several checkboxes and input fields:

- Replace Scripted Item Quantity: [1]
- Component Min Items: [5] Fixed Quantity
- Roll Up Cost Max Items: [5] Prompt For Qty
- out of stock only

At the bottom are buttons for "< Ok >" and "< Cancel >". The window title bar at the bottom right says "Script Editor".

Add Item from List Script

If the Replace Scripted Item parameter is marked ON, then when the Scripted Item is specified on a sales transaction, it is automatically replaced by an alternate selected item. If this parameter is marked OFF, both the Scripted Item and the selected item(s) are entered on the transaction.

Component

The Component parameter may be used to mark a selected item as a component of the Scripted Item rather than as an additional sales item, (e.g., for the purpose of kitting or sub-assembly). The term "selected item" refers to supplementary items added using either a Tag-along Item or an Add Item from List/Range Script.

Component items are the same as normal line items, except they do not possess a selling Price and they may be printed differently on the sales receipt or Invoice. Although Component Items do not have a selling Price (by definition), they may carry a Cost, unless the Roll Up Cost parameter is turned ON.

If the Component parameter is marked ON, any selected items are marked as components. Component items are added to the sales transaction immediately following the Scripted Item and the price will be omitted (it is actually set to \$0.00 and not displayed on the screen). If the Component parameter is marked OFF, selected items are added to the sales transaction the same as normal line items in every respect.

How Component items are printed are dependent on how your form is designed. Depending on the Form File being used, Component items may be printed the same as normal line items, in a different typeface (e.g., condensed), or not at all. If you would like to change how Component items are printed, please contact your local Authorized **Keystroke** dealer for a custom modification to your Form File.

Note: Once a Script has been completed, you can manually enter a sale Price for a Component item on the sales transaction screen. When this is done the item is no longer considered a Component, and automatically becomes a standard line item.

Roll Up Cost

The Roll Up Cost parameter provides the ability to have the Cost and/or selling Price of selected items consolidated into the Cost and/or Price of the Scripted Item. The purpose of this parameter is to control how Costs and Prices will be reflected on profit reports. The Roll Up Cost parameter is available on the following Script Types: Tag-along Item, Add Item from List and Add Item from Range.

If the Roll Up Cost parameter is marked ON, the Cost and selling Price of selected items will be automatically added to the Cost and Price of the Scripted Item. However, if the Component parameter is also marked ON, selected items have no Price; therefore, only the Cost is rolled up.

Note: If the Roll Up Cost parameter is used and you edit, delete or change selected items for which the Cost and/or Price have been rolled up, the program will not readjust the Cost or Price of the Scripted Item (the original rolled up Cost/Price will not change).

Out of Stock Only

The Out of Stock Only parameter provides the ability to make an individual Script function only when the Scripted Item is not in stock. Out of Stock Only Scripts are *not* executed when the Scripted Item is in stock. This parameter is available to be used on all types of Scripts.

If the Out of Stock Only parameter is marked ON, the Script will function only when the Quantity On Hand of the Scripted Item is 0 (zero) or less. If this parameter is marked OFF, the first Script will function only when the Scripted Item is in stock.

For example, an item has some Scripts with this parameter ON and some Scripts with the parameter marked OFF. When the item is out of stock, the only Scripts that will run are those with Out of Stock Only marked ON. Conversely, when the item is in stock, the only Scripts that will run are those with Out of Stock Only marked OFF.

If a Scripted Item consists of no Out of Stock Only Scripts, all Scripts will run regardless of whether or not the item is in stock. However, if the Clerk's Security Level does not allow him or her to sell items that are out of stock, no Scripts will be initiated, regardless of how this parameter is set.

Quantity

The Quantity field is used to specify the default sale quantity of a selected item per Scripted Item. How this value is used is dependent on the Fixed Quantity parameter described below. In other words, it can represent either a fixed quantity or a multiple quantity of the Scripted Item. The Quantity parameter is available on the following Script Types: Tag-along Item, Add Item from List and Add Item from Range.

Repeat Items

The Repeat Items field is available only on a Tag-along Script, and is used to specify the number of times to repeat the entry of the selected item per Scripted Item. If you specify a Quantity of 2, and a Repeat Items value of 4, a total quantity of 8 of the selected item will be added to the sales transaction (4 lines of 2 items each). The default Repeat Items setting on a Tag-along Script is 0 (zero) or 1, which will cause the Tag-along Item to be entered only once.

Min Items

The Min Items parameter may be used to require a minimum number of items to be selected. The Clerk cannot leave the list of items via the **[ESC]** key until the minimum number of items has been selected. For example, if the Scripted Item is a gift basket which consists of one basket and several gift items, the Script for the basket may provide a list of several types of baskets and require that at least one be selected (the Min Items for that Script would be set to 1). If the gift basket always contains 2 packages of gourmet coffee, the Min Items for that Script would be set to 2, and would require the Clerk to select 2 packages of coffee.

Max Items

The Max Items parameter may be used to limit the number of items that may be selected in an Add Item Script. In other words, once the maximum number of items have been selected, the Script will automatically leave the list of items. In the example given above for the Min Items parameter, a subsequent Script for the gift items might have the Max Items set to 5, so the Script would end after five gift items have been selected. The Max Items for the basket Script might also be set to 1, so the Clerk is prompted to enter only one basket.

On Tag-along Scripts, this setting is labeled Repeat Items since it can then be used to automatically repeat the same line item on the transaction. The default Repeat Items setting on a Tag-along Script is 0 (zero) or 1, which will cause the Tag-along Item to be entered only once.

Note: If the Replace Scripted Item parameter is turned ON, the Max Items parameter should normally be set to 1. Otherwise the Clerk will be encouraged to select more than one alternate item, when there is only one item to be replaced.

Fixed Quantity

The Fixed Quantity parameter may be used to set the quantity of a selected item to a specific value (see Quantity, above) regardless of the sales quantity of the Scripted Item. This parameter is available on the following Script Types: Tag-along Item, Add Item from List and Add Item from Range.

If the Fixed Quantity parameter is marked ON, the sale quantity of the selected item will be equal to the Quantity setting. For example, you might use a Script to add “delivery” as a Tag-along Item, in which case you would probably want a quantity of 1, regardless of how many items are being delivered. Or, as a sales promotion you might offer a “free product sample pack” (limit one per customer) with the purchase of a particular item.

If the Fixed Quantity parameter is marked OFF, the sale quantity of a selected item will be equal to the sale quantity of the Scripted Item times the Quantity setting. For example, if you’re selling 5 of the Scripted Item, and the Script Quantity is 2, a Tag-along Item would be entered with a sale quantity of 10; if Fixed Quantity is marked ON, the sale quantity of the Tag-along Item would be 2.

Note: The sale quantity of a Tag-along Item (or selected item(s) via the Add Item from List/Range Types) can be manually changed once a Script has been completed.

Prompt For Quantity

The Prompt For Quantity parameter is used to allow the sales Clerk to specify the sale quantity of a Tag-along or selected item. If this parameter is turned ON, the Script will prompt you to specify the Quantity for the selected item. For example, if you sell a Scripted Item for 5 T-shirts using the “Add Item from List” Script type. The Clerk could then select Large T-shirts, quantity of 3; and Small T-shirts, quantity of 2.

Script Editor Menus

The Script Editor menu bar includes the Common menu options, along with the following menus and functions:

Script Menu

- Clear** clear data from the pending Script setup screen
- Edit** edit a previously saved Script
- Delete** remove a previously saved Script
- Recall** restore a Deleted Script
- Save** save the currently displayed Script(s)

Reports Menu

- Scripts** print a listing of existing Scripts


Special Menu

- Copy** copy displayed Scripts to memory buffer
- Paste** add Scripts from buffer to current Script list
- Pack** rebuild Script data files
- Export** export Scripts to an ASCII delimited file
- Import** import Scripts from an ASCII delimited file

Script Menu

The functions on the Script Menu are used to manage individual Script records.

Clear Function

The Clear function removes all information currently displayed on the Scripted Item dialog box. Any changes that have been made to the current Scripted Item will be discarded. The Clear function does not affect any Scripts or settings that have previously been saved. You can also press  to Clear the Scripted Item dialog box.

Edit Function

The Edit function is one way of bringing up a Scripted Item in order to change its settings, modify its Scripts, or to test its operation. When you select the Edit function, specify how you wish the list box of Scripted Items to be sorted by selecting Description, Product Code, Stock Number or Class. A list box showing any Scripted Items will be displayed, from which you can select the Scripted Item you wish to bring up.

Note: It's actually faster and easier to just enter the desired item on the Scripted Item dialog box or press **F2** to select the item from a list box. This list box, however, will show all Inventory items, regardless of whether they have Scripts attached to them. The list box accessed from the Edit function will show Scripted Items only.

Delete Function

The Delete function is used to remove all Scripts for an item. When you select the Delete function, first specify how you wish the list box of Scripted Items to be sorted by selecting Description, Product Code, Stock Number or Class. A list box showing Scripted Items will be displayed. Highlight the item for which you wish to delete the Script and press **ENTER**.

Recall Function

The Recall function does the opposite of what the Delete function does. When you select the Recall function, a list box will be displayed showing Scripted Items that have been deleted (since the Script files were last Packed). To recall an item, highlight it and press **ENTER**. If no Scripted Items have been deleted, or if the Script files have been Packed since the Script Delete function was last used, the system will give a warning beep to indicate nothing is available to be recalled.

Save Function

The Save function records the Scripts as currently displayed on the Scripted Item dialog box. A Scripted Item can also be saved by pressing **F10** or selecting the < **Save** > button on the Scripted Item dialog box.

Reports Menu

The Reports menu is used to access the Scripts reporting function.

Scripts Function

This function is used to print a listing of Scripted Items and their respective Scripts. The settings for the Scripts Report dialog box are nearly identical to those used throughout the standard Report Manager module. For instructions on how to use the standard report settings, please refer to the Report Manager section of this User Guide.


The < **Detail** > button on the Script Report dialog box is used to select the level of detail the report will include. If you specify Scripted Items, the report will list only the items that are Scripted but will not show the scripts themselves. The Scripts setting includes each Script and its respective parameter settings. The Detailed report option shows all details for your Scripted Items, including Message, Individual Item(s), and/or item Range parameters.

Special Menu


The Special Menu includes the Copy, Paste, Pack, Export, and Import Script functions.

Copy Function

The Copy and Paste functions are used together to allow you to easily copy Scripts from one Scripted Item to another. Use the Copy function to copy all the Scripts from the current Scripted Item into the computer's memory; then use the Paste function to append those Scripts onto a different item.

To use the Copy function, first bring up the Scripted Item from which you wish to copy Scripts. Then either press  or select the Copy function from the Special menu. The Scripts from the currently displayed Scripted Item will be copied into memory. See the Paste function below for directions for the remainder of the procedure.

Paste Function

The Paste function is used to add the Scripts specified using the Copy function to the item displayed on the Scripted Item dialog box. To use this function, you must first use the Copy function to specify which set of Scripts to copy, then whenever you press the  key (or select the Paste function from the menu), the copied Scripts are appended to (added to the end of) the list of Scripts on the Scripted Item dialog box.

Pack Function

The Pack Scripts function is used to rebuild the data files in which your Script records are stored. This permanently removes deleted Script records and can repair damaged data files (depending on the type and severity of the damage). Generally, it is not necessary to Pack Scripts unless there is a problem with the Script data files; however, there are two reasons that may warrant the use of this function.

One reason is when a significant number of Script records have been deleted. Packing permanently removes all deleted Script records and rebuilds the indexes. In theory, this should make the Script data files smaller and faster; however, it is doubtful this would make a noticeable difference. Another reason you may need to Pack Scripts would be to attempt to recover damaged Script data files, (e.g., perhaps caused by a failing hard disk). In such a situation, however, it may be more logical to restore an undamaged copy of the files from an earlier backup. Otherwise, these functions should not be used unless recommended by **Keystroke** Technical Support.

Export Function

The Import and Export functions can be used to transfer and/or update Scripts from one set of **Keystroke** data files to another (e.g., from one store to another).

The Script Export function on the Special menu is designed to be used to transfer Script data to a standard ASCII delimited text file which can in turn be Imported into another set of **Keystroke** data files. The settings on the Export function include < **Sort By** >, < **Filter** >, and < **File Name** >, and are essentially the same as comparable Report settings.

To Export all Scripts for the purpose of transferring Script data to another **Keystroke** system, the suggested settings are: < **Sort By** > – Stock Number, < **Filter** > – None, and < **File Name** > – EXPORT.TXT.

Import Function

The Script Import function on the Special menu can be used to transfer and/or update Script data from an ASCII delimited text file generated by the Script Export function described above. The settings on the Import function include File Name, < **Search By** >, Add New Scripts, and Update Existing Scripts.

To Import Script data, enter the appropriate File Name (same as specified in the Script Export function) and verify each of the other Import settings. The recommended < **Search By** > setting is Stock Number, and one or both of the toggle settings must be marked (Add New Scripts and/or Update Existing Scripts).

Managing Script Data Files

All information regarding Scripts are stored in four data files; **KSSCR.DAT**, **KSSCRITM.DAT**, **KSSCRNDX.DAT**, and **KSSCRCMT.DAT** (i.e., **KSSCR*.DAT**). These files should be backed up regularly, along with all other **Keystroke** data files.

There are some cases (other than regular backups) where it may be necessary to copy or transfer the Script data files independently from other **Keystroke** data files. For example, if you have more than one store running the program with identical InventoryI records, it would be a great time-saver to set up Scripts in one store and copy the resulting Script data files to subsequent stores.

Whenever Script data files – or any **Keystroke** data files for that matter – are to be transferred, it is important that you transfer all four of the files listed above. Missing or mismatched files from this set will generally lead to undesirable results.

Importer Module

The Importer module is used to transfer Customer, Vendor and Inventory information from other programs into the **Keystroke** system. Functions are provided to import entire database records, or to merge only specific information (such as updated prices) into existing records. Data to be imported must be in standard ASCII delimited (quote/comma or tab) file format and must satisfy the guidelines described below.

The task of importing database records may also be integrated into automated processing programs for unattended operation (i.e., batch programs or communication scripts) by running the Import module (file name IMP.EXE) from the DOS prompt. See the Importer topic in the Direct Access from DOS Prompt section of the Advanced Features Appendix for more information.

Source Data Guidelines

The term “Source” data refers to data and data files that are to be imported. The Source data must satisfy specific guidelines in order to be successfully imported into **Keystroke**. If the Source data does not strictly adhere to these guidelines, errors may result and **Keystroke** data files may be damaged. Therefore, it is highly recommended to always make a backup copy of **Keystroke** data files prior to importing data.

The Source data files must be in standard ASCII quote/comma or tab delimited format. In other words, the files must contain only ASCII characters (except those noted below), and each field **MUST** be separated by either a comma or a tab character. ASCII characters are generally those characters that are available on a standard PC keyboard. A “field” refers to each element of information, such as a Product Number, Description, Date or Price. When using comma delimited files, each field should also be delimited by Double Quotation marks (“ , ”) placed at the beginning and end of each field.

The Tilde (~) character may not exist anywhere in the Source data file. On comma delimited files, the Comma (,) character may be used in data fields *only* if the field is delimited with Double Quotation marks as described above. Any Comma that is not enclosed within Double Quotation marks will be interpreted as delimiting a new field. The name of the file to be imported must conform to DOS file name format (maximum eight character prefix plus maximum three character extension, separated by a period; e.g., **IMPORT .TXT**).

Importer Menus

The Importer Module menu bar includes the Common menu options, along with the following menus and functions:

Import Menu

Inventory import Inventory data
Customers import Customer data
Vendors import Vendor data

Setup Menu

Inventory configure for Inventory data import
Customers configure for Customer data import
Vendors configure for Vendor data import

Setup Menu



The functions on the Setup menu are used to define parameters for the Import functions:

Inventory Set up Inventory Import parameters
Customers Set up Customer Import parameters
Vendors Set up Vendor Import parameters

Before using the Importer module to transfer data, various parameters such as Source data file names and field arrangements must be specified in the appropriate Setup function. A separate Setup function exists for each type of database record that may be imported: Inventory, Customers, and Vendors.

From the Setup menu, select Inventory, Customers or Vendors, depending on the information you wish to import. A dialog box will display the current Import Setup parameters.

Setup File:

All Import Setup parameters are stored in a Setup File. To view a list of available Setup Files, select the <Setup File> button. You may then select an existing Setup File, or press  to create a new Setup File. If you press , a dialog box will appear in which you may enter a unique file name for your new Import Setup File. All Setup Files will automatically be assigned the extension “.KSI”.

Import File:

The Import File field is used to specify the DOS file name and/or path where the Source data is stored, (e.g., A:\CUSTOMER.TXT). Below the Import File field on the Setup screen is a listing of all data fields or variables that may be imported for Inventory, Customer or Vendor database records. The column to the right of each field variable represents the position of the data field as it is stored in the Source file. For each field that is to be imported, enter the appropriate number for the field position next to the corresponding **Keystroke** data field. The field position should be left blank or 0 (zero) for field variables that are not to be imported or are not available in the Source data file.

First Record/Last Record

The First Record and Last Record fields are used to specify the range of records which are to be imported. To import all records stored in the Source data file, the First Record should be set to **1** and the Last Record should be left blank or **0** (zero). Otherwise, specify the desired First and Last record number in the appropriate fields. (Some source files may contain a first record which is blank or contains the titles of the fields; in this case, set the First record field to **2**.)

< Merge By >

The <Merge By>, Add New Items, and Don't Merge Existing Items parameters pertain only to the function of merging updated information (such as price updates) into existing database records. These parameters will have no affect when using the Append or Replace functions.

The <Merge By> parameter is used to specify which field to look up records by in order to determine a matching record when using the Merge function (e.g., for Inventory: None, Description, Product Code, Stock Number, Class or Manufacturer). If a <Merge By> field is not specified, the Merge function will not operate properly. If the Merge function is to be used, the field being merged by must be imported (its field position on the list must not be 0 (zero)).

Add New Items

The Add New Items parameter determines whether new items (items in the Source data file that are not found in the current database) should be imported when using the Merge function. If this parameter is marked OFF or blank, the Merge function will ignore records in the Source file that do not match an existing record based on the selected <Merge By> field.

Don't Merge Existing Items

The Don't Merge Existing Items parameter turns off the updating of items already in the database and will cause the program to only add new items to the database.

Note: Do not turn this parameter ON if Add New Items is OFF. Doing so will cause the Importer to spend a lot of time doing nothing.

After specifying all necessary Import Setup parameters, select the **[Ok]** button or press **F10** to save the Importer setup information.

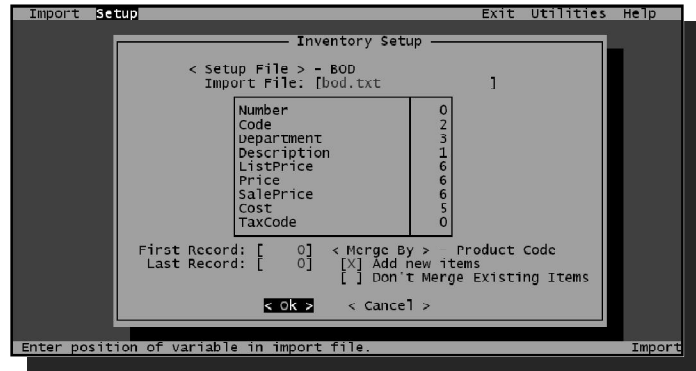
Example:

Below is an example of a Source data file which contains basic Inventory information (product description, part number, cost, price, etc.) for two Inventory items.

```
"MOUNTAIN BIKE - RED", "GTMB12345", "MENS", "A",  
  "250.00", "495.95"  
"BICYCLE ODOMETER/SPEEDOM", "OD987654", "ACC", "B",  
  "12.50", "29.95"
```

For this Source data, the Import Setup parameters might be entered as shown in the dialog box below. Note that several of the data fields are marked with a 0 (zero) in the position column. This means that the field is not available to be imported and will either be left blank, assigned a default value (as defined in the Setup Default function in the Database Manager), or assigned a sequential number (Stock Number, Customer Number or Vendor Number fields only). In most cases, it is best to allow the system to assign the Stock Number, Customer Number and Vendor Number.

This screen indicates the proper setup for importing the preceding example data.



Inventory Import Setup screen

Note that several of the variables are marked with the same position, such as the Cost and Price fields (positions 5 and 6). In most cases, if the Source data file contains only one Cost or Price per item, those fields should be imported into every appropriate field. Similarly, if you have only one Address, City, State and Zip per Customer, each of those fields should be imported twice, (once for the Billing Address and again for the Shipping Address). Conversely, you might note that field position 4 is not used; not all fields in the Source data file must be imported.

Import Menu

The functions on the Import menu are used to activate data import operations:

Inventory Merge, Append or Replace Inventory data

Customers Merge, Append or Replace Customer data

Vendors Merge, Append or Replace Vendor data

After completing the appropriate Setup functions, select the type of records you wish to import from the Import menu (Inventory, Customers or Vendors). Next, select Merge, Append or Replace, depending on which operation you wish to perform.

The Merge option is used to import modified information (such as price updates) into existing database records. The Append option is used to add new records to an existing database. The Replace option is used to delete the contents of the existing Inventory, Customer or Vendor database and import new data.

Before beginning any of the Import operations, it is highly recommended that you make a backup copy of all data files that may be affected. The **Keystroke** data files that are affected by the Import module are: **KSINV* .DAT** for Inventory, **KSCST* .DAT** for Customers, and **KSVND* .DAT** for Vendors.

Once an Import operation is begun, you may press “**X**” at any time to pause or quit. Any data imported before halting the Import operation will be saved in the **Keystroke** data file.

If an error is encountered, refer to the Troubleshooting Tips at the end of this section for more information. In most cases, any errors encountered during an Import operation are caused either by an incorrect Setup parameter or a violation of the Import Source Data Guidelines (discussed earlier).

After Importing Data

Before continuing with operations in the **Keystroke** program that depend on this data, all imported data should be thoroughly reviewed in order to verify the accuracy of the data transfer. Especially with large data files, it is very possible that the Source data is not 100% consistent with the guidelines of the Import function.

We suggest you begin by printing out detailed listings of all imported data using the functions in the **Keystroke** Report Manager. A more complete method of verifying the data is to review each imported Database record in the Database Manager.

Data Field Specifications

All fields (except for the special fields noted below) are listed in the same order that they would be exported using the Export function in the Configuration Manager. If you are importing data that was exported from **Keystroke**, you can quickly go down the list and simply enter field numbers in the order that they appear (i.e., start with 1, then 2, and continue to the end of the list). Remember to leave the special fields (those with an *****) with a setting of **0** (zero).

A list of the data fields and what type of data they may contain is located in the file EXPORT.DOC, located in the DOC\ subdirectory. You can view this file using any text editor.

The following fields are not standard **Keystroke** fields, but are available to be imported because other programs may store their data in that format. These special fields are denoted by an *****:

- * Phone
- * Bill City+ST [+ZIP]
- * Ship City+ST [+ZIP]
- * First + Last Name

These special fields may be used to import information stored as a single field in the Source data file into multiple fields in **Keystroke**. The Importer program will interpret various formats for each of these fields (e.g., the Phone field may be stored as either “**505-123-4567**” or “**(505) 123-4567**”). The Mail/Ship City+ST [+ZIP] fields may be “**Long Beach, CA 90803**” or “**Long Beach, CA**”. The First + Last Name field may be stored as “**Bob Jones**” or “**Bob D. Jones**”).

Troubleshooting Tips

Most errors are caused by improper characters in the file being imported. The most common of these problems is caused by having quote characters (used as inch marks) within the description field. Since a quote/comma delimited file uses quotes to mark the beginning and ending of a field, stray quotes or commas can cause fields to be imported incorrectly. For more information, refer to the section on Error/Warning Messages located at the end of this Appendix.

Production Module

The **Keystroke** Production module is used to manage Inventory costs and quantities for items which are assembled and/or sold as a single product, but purchased as separate components. The Production module may also be used to break down products which are purchased in bulk and sold as smaller units.

Similar functions can also be accomplished using the Custom Script Editor module, as well as using the Parent/Child and Order Multiplier features which are integrated into the Inventory database. However, the primary difference between these methods and using the Production module is that these methods are real-time and more integrated, while Productions are performed on a batch level and must be entered as transactions in order to affect Inventory.

The Production Module menu bar includes the Common menu options, along with the following menus and functions:

Production Menu

- Clear** clear data from the pending Production screen
- Edit** edit a previously saved Production
- Delete** remove a previously saved Production
- Recall** restore a Deleted Production
- Date** change the date/time of the current Production
- Complete** go to the Save Production box

Template Menu

- Select** select an existing Production Template
- Auto Select** select the next Automatic Production Template
- Auto Produce** run all Automatic Production Templates
- Delete** remove a previously saved Production Template
- Recall** restore a Deleted Production Template

Reports Menu

- Productions** generate a listing of Productions
- Templates** generate a listing of Production Templates

File Menu

- Pack Productions** rebuild Production data files
- Pack Templates** rebuild Template data files

Introduction

The Production module is an InventoryI maintenance tool designed with two general purposes in mind. Its primary purpose is to provide the ability to adjust Inventory levels for items that are produced from individually purchased components and sold as a finished product; hence the name Production. An example of a Production item is a computer which may be purchased as a case, mother board, power supply, disk drives, etc., and is sold as a single finished product. The second purpose of the Production module is to provide the ability to manage Inventory levels of products that are purchased in bulk units and sold individually. This operation may be referred to as a “Reverse Production” or “Case Break-Down”.

Note: The Parent/Child feature on Inventory items provides the same Case Break-Down feature and has the advantages of being easier to maintain and of operating in real-time (the cases are broken down as you use them) instead of in batch mode. Production is a batch mode process, which means that nothing happens until you actually tell the program to perform the operation, typically at the end or beginning of each day.

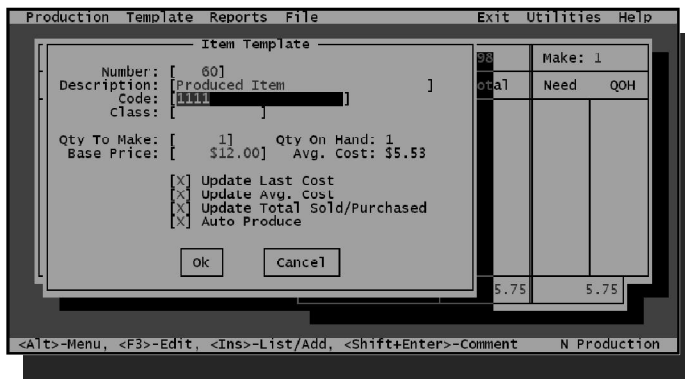
Using the Production Module

Whether an item is being produced from components or a case is being broken down, the operation is referred to as a Production run. When a Production run is saved, Inventory levels and costs are adjusted according to the quantities that are requested. All aspects of a Production run (component items, costs, quantities, etc.) may also be saved as a Template that can be reused for subsequent runs of the same Production item.

To generate a Production run, first specify the item to be produced (or broken down) on the Item Template dialog box. If a Template has previously been saved for that item, the standard components and other settings for that Production item will be displayed.

On the Item Template dialog box, specify the Quantity To Make, verify the Base Price, and mark whether Last Cost and/or Average Cost should be updated. For convenience, the current Quantity On Hand and Average Cost are also displayed in this box.

After completing the Item Template box, press **F10** and specify the component items which make up the item being produced. The **F8** key will take you back to the Item Template box. Items may be entered by Stock Number, Product Code, Description, Class or Manufacturer. The default entry mode is determined from the Parameter settings in the Purchase Manager module. The **F9** key will toggle between the four item entry modes.



Production Template

Verify the required Quantity of each component, as well as the Cost of each component. The Qty, Cost, and Total columns are based on the Production of one item. The total cost of all components is displayed at the bottom of the Total column. The current Quantity On Hand (QOH) of each component item and Need Quantity for the current Production run is shown to the right of the double vertical line. If the QOH is less than the Need Quantity of any component, that item is marked with a “b” in the QOH column. This mark is for your attention only and will not affect the Production run.

After entering all necessary information for the Production, press **F10** or **ENTER** on the last line. Four buttons will then be displayed. Select the **< Optional Items >** button to add optional items to the Production run, the **< Save Template >** button to save a copy of the Production as a Template for later use, and/or the **< Save Production >** button to save the current run and adjust the appropriate Inventory levels and costs. Select **< Clear >** to abandon the Production run and return to a blank Item Template box.

Keystroke

Reverse Productions (case breakdowns). The fundamental difference between these two applications is whether the Quantity To Make is greater than or less than zero. The other

Productions from Components

For normal Productions the Quantity To Make will be a positive number. When the quantity is increased by the specified Quantity To Make, and the Quantity On Hand of each of the components will be decreased by the Need quantity. If the Update Last Cost and Update Cost options are selected, the program will automatically adjust the costs on the Inventory record of the item being produced. The cost adjustments are based on the total cost of the component items.

For Reverse Productions, the Quantity To Make is negative. When the Reverse Production is saved, the Quantity On Hand of the item being broken down will be decreased by the quantity. The Quantity On Hand of the component items will be increased by the Need quantity.

Cost Adjustments


with normal Production runs. Because the component item is what is being produced, any necessary cost adjustments will be made on the Inventory records of the component items.

The parameters on the Item Template box are turned ON.

As each component item is entered, a cost will be displayed for that component. The cost

Therefore, as items are entered, the program will attempt to adjust the component costs in order to match the cost of the item being broken down. If the components are a multiple

component is determined by dividing the Average Cost of each case by the quantity per case. The cost of each component will be rounded to an amount that is as close as

If the item being broken down consists of more than one component, the cost of each will be initially set to the Average Cost of the component. After entering each of the component costs, press  on the last line. The program will automatically compare the

If these amounts are not equal, a dialog box will appear, giving you several options on how to proceed. If you select **Yes**, the program will calculate component costs based on the relative Average Cost of each component in order to equal (or come as close as

possible) to the total cost of the item being broken down. Pressing **<No>** will return you to the main screen, where you can manually adjust the component costs.

Pressing **<Cancel >** or pressing **ESC** will return you to the main screen.


Production Menu


The functions on the Production menu are nearly identical to those found on the Invoice menu in the Sales Manager: Edit, Delete, Recall, Date, and Complete. The Clear function is used to remove all information currently displayed on the Production screen. This does not affect Productions or other data that has already been saved. The Edit, Delete and Recall functions are identical to those found elsewhere in the **Keystroke** program. Whenever these functions are used, Inventory levels will be adjusted accordingly. The Date function is used to specify a date other than the current system date for the Production run.

Template Menu

The Select and Auto Select functions on the Template menu are used to specify the Template which will be used for the next Production run. The Auto Produce function is used to automatically produce all items which need to be produced based on the current Quantity On Hand and Minimum Quantity levels. The Delete function is used to remove a Template that has been previously saved, and the Recall function is used to Recall a Template that has been previously Deleted.

Select & Auto Select Function

The Select function is used to choose a Production Template from a list box of previously saved Templates. The Auto Select function is very similar to the Auto Order function in the Purchase Manager module. Choosing this function or pressing  will automatically search existing Templates for items that need to be produced based on the current Quantity On Hand and Minimum Quantity levels of those items.

The Auto Select function checks each Template in order of the current item entry mode (Stock Number, Product Code, Class, or Description). When an item is located that needs to be produced, the Template for that item will be selected and you can complete the Production run for the item, if desired. Press  again to search for the next item.

Auto Produce Function

The Auto Produce function is used to automatically produce all items which need to be produced based on the current Quantity On Hand and Minimum Quantity levels. When the Auto Produce function is selected, the program will immediately begin searching Templates for items that need to be produced. When a Template is located for an item with a total Quantity On Hand (plus On Order) which is less than the Minimum Quantity, a Production run will be automatically generated.

Items will only be automatically produced if the Auto Produce parameter is marked ON on the Production Template for that item. The Quantity To Make will be based on the Qty To Order. The costs will be automatically calculated and adjusted using the current average costs from the Inventory database.

This function will pre-evaluate items which are setup for reverse Productions (Case Break-Downs) in order to determine proper quantities to be produced. This is only applicable to chained item Productions (i.e.; case to 12-pack, 12-pack to 6-pack, 6-pack to singles).

Note: The Auto Produce function can also be run directly from the DOS prompt. This allows you to include it in a batch or scheduled program which is to be automatically run at a predetermined time. Please see Auto-Production from DOS Prompt in the Advanced Features section of this Appendix for more information.

Production Templates are a convenient method of recording Production criteria for items that are frequently produced, thus eliminating the need to re-enter the same information each time an item is to be produced. All aspects of a Production run (Component Items, Quantities, Costs, etc.) may be saved as a Template. Templates may also include optional items that can be selected for individual Production runs.

To create a Template, follow the same general procedure that is used to enter a Production run. Because the Template is actually an inherent part of a Production run, the Template can be saved at the same time as a Production run by simply selecting the [Save Template] button before saving the Production.

Optional Items

A Production Template may include one or more Optional Items which can be easily selected for each Production run. To access the Optional Items entry box, select the <Optional Items> button on the save box. The Optional Items entry box operates in two modes: one for selecting items to be attached to the current Production run, and another for editing the Optional Items which are to be stored with the Template.

When you first access the Optional Items entry box, the program will be in Select mode. While in Select mode, you may press **ENTER** to attach an Optional Item to the current Production run, or **DEL** to remove an item from the list of Optional Items for the Template. To change the mode of operation to Edit mode, press the **F3** key. In Edit mode you may add or remove Optional Items from the Template by following the same procedures used elsewhere in the program (i.e., **INS** for an Inventory list box, **CTRL** **BkSp** to remove an item). To change back to Select mode from Edit mode, press the **ESC** key. To exit the Optional Items function, press **ESC** while in Select mode.

Comments

Comments may be attached to Production runs or Templates by pressing **SHIFT** **ENTER**. If a Comment is saved with a Template, that Comment will be copied to each Production run on which the Template is used.

Reports Menu

The Reports menu consists of functions for generating reports listing Production runs and Templates. These functions operate the same as those in the Report Manager module.

Productions used to generate a report of Productions run during a specific Date/Time range

Templates used to generate a report of Templates created for Production runs

File Menu

The File menu consists of two file maintenance functions: Pack Productions and Pack Templates. These functions are available for the purposes of permanently removing deleted records and repairing damaged data files. It is usually not necessary to use either of these functions.

There are generally two situations in which the File Pack functions are useful. One case is when a significant number of records (Productions or Templates) have been deleted. Packing will permanently remove those deleted records and rebuild the indexes which are used to look up records.

The other situation would be in the event of a power failure while a Production or Template is being saved, which might result in damage to the file. Otherwise, these functions should not be used unless recommended by **Keystroke** Technical Support.

Advanced Features Appendix

The Advanced Features section of this User Guide includes a wealth of valuable information that is not necessarily required reading for the typical **Keystroke** user. Many of the topics in this Appendix should normally be addressed when the system is first installed (typically by an Authorized **Keystroke** dealer).

Along with a variety of other topics, the Advanced Features Appendix includes instructions for running **Keystroke** on multiple machines using a local area network (“LAN”), descriptions of all data files used by **Keystroke** and how to manage them, and instructions on using the numerous stand-alone utility programs which come with the **Keystroke** program.

Protecting Your Investment

Your investment in **Keystroke** lies not only in the cost of hardware and software, but even more so in the value of the data you’ve collected by using the hardware and software. To put it into perspective, estimate how many hours it would take you and your employees to re-enter all your data into a new computer system. Then calculate how many dollars that time would cost, and compare that amount to what you paid for all the computer hardware and software you own.

Now that you understand the value of your data, you would probably like to take out an insurance policy on it. Unfortunately, there’s no insurance company in existence who can pay any amount of money to replace your data if it is lost; however, there are several relatively inexpensive methods of self-insuring your investment:

- Make backup copies of your program and data files on a regularly scheduled basis
- Feed only clean power to your computer(s)
- Prepare for equipment failures

These are the topics covered in this section, making it perhaps the most important section in this User Guide. While there is not much anyone can do to force you to regularly back up your files, please DO NOT consider this an optional task.

Backing Up Program and Data Files

You should back up your data files every day. This should be a part of your daily closing procedures, ranking right up there with securing the cash and locking the doors. Computers are wonderful machines, but they can still break down, be stolen, or destroyed in a disaster. These are just a few of the reasons why it is extremely important for you to make regular backups.

In the event of a system failure, having a complete backup copy of your data files can be like having a good spare tire when you get a flat on a desert highway. The ability to restore your valuable data can be a tremendous savior. At the same time, consider the consequences of unexpectedly losing all the valuable information regarding your Inventory, Customers, and sales. Everything you've entered into the **Keystroke** program over the past months or years could be damaged or destroyed by a system/hardware failure. This can happen, and does happen. Please assume that it will happen to you, and make daily backups!

To get started, you'll first need to have plenty of diskettes on which to store your backup files. The type and number of disks you'll need depends on how much data you have and what type of removable disk drive is available on your machine. Below is some general information that should help you determine what type of disks you need.

Your machine should have at least one floppy drive that takes either 5¼ inch or 3½ inch disks. Unless it is a fairly old machine, these drives will probably take High Density disks. HD 5¼ inch disks have a capacity of 1.2 Mb (Megabytes, 1,213,952 bytes), and HD 3½ inch disks have a capacity of 1.4 Mb (1,457,664 bytes). Another newer disk format is called the Zip drive from Iomega®, which has a capacity of up to 100 Mb using special Zip disks. If you choose to use a tape backup system, you will need to use a third party backup software which supports the tape backup device; the Backup function in **Keystroke** does not support tape backup devices.

You can purchase disks from your local **Keystroke** dealer, who can also help you learn how to make reliable backups. If you have absolutely no idea how many disks you might need, ask your dealer and/or purchase plenty; you will use them eventually. The Standard Backup function will compress **Keystroke** data files to about 20% of their actual size. 10 Mb of data files plus 3 Mb of program files should easily fit on 4-5 high density floppy disks.

Each time you make a backup, write the date it was created on the first disk. It's OK to reuse the same disks to make a newer backup; however, *do not* use the same set of disks every day. Periodically, such as once a week or month, take a set of backup disks home with you. They're not that expensive, and you never know when they might come in handy. There are several theories on this, but the most important idea is that you should always have several sets of complete sets of backup disks on hand, each made on a different date.

For example, back up your data files each day of the week, labeling them Monday through Friday. At the end of each week (or month) replace one set of disks with new ones, and put that set of disks away. Each time you back up, rotate the day of the week that you replace so you'll always be using relatively new disks.

Another important procedure is to periodically check to see that the backups you are making are good and usable. Although it may appear that your backups are completing properly, the only way to be absolutely sure is to restore the files onto a different machine, and test them by running the program and reviewing the actual data (e.g., run several reports). Again, your local **Keystroke** dealer can help you with this.

Note: For maximum safety, use a third party backup software product such as PC-Tools®, Norton Utilities®, FastBack®, or DOS-MSBackup®. Because these products are specifically designed for the purpose of backing up files, they can generally be expected to perform superior error checking and provide improved speed, flexibility and reliability over the simplified backup utilities built into the **Keystroke** program.

Power Conditioning / Battery Backup

Ensuring that your computers are getting good, clean, and reliable power is one of the most important and economical things you can do to protect yourself from computer malfunctions and data loss. Many studies have been done by some of the largest computer technology companies (including IBM, Bell Laboratories, and several others) and while the numbers vary, they all have found that about 80% of all computer problems are caused by bad power. They also learned that 80% of all power problems are power sags or brownouts. The problem is that surge protectors (even expensive ones with line/noise filters) can't do anything about low power (low voltage) conditions. While they do protect your computer system from being "fried" by a power spike or surge, they cannot ensure that it will run properly during a sag or brownout.

What your computer really needs is a power protection device that provides true Voltage Regulation (also called Line Conditioning). There are two types of these devices available, those with Battery Backup power (Uninterruptable Power Supply or "UPS"), and those without, which are typically called Line Conditioners (no battery backup, so when the power goes off so does your computer).

A Battery Backup (or UPS) device contains a battery that can supply enough power to run your computer for several minutes after the electricity has gone out. These devices come in a wide array of features (and prices). The more expensive ones usually provide better line conditioning and larger battery capacities to allow you to run the computer for longer periods of time. Battery Backup devices, which cost from \$150 to \$300, should provide fairly good power conditioning and will allow you to continue to run your computer from 5 to 20 minutes after the power goes out.

Be careful of some of the less expensive "Stand By" type UPS devices. Some of them do not have very good (if any) power conditioning capabilities. They are designed to provide emergency power in the event of a power failure, but not necessarily good clean power, which is the most likely cause of malfunctions and data corruption. While they will keep your computer running during a blackout, they may not help it during a power sag (which is when most problems actually occur). Please consult with your local **Keystroke** dealer for their recommendations.

Fail-Safe Network:

How to Stay Open When Your Network Goes Down

There actually is no such thing as a fail-safe network; however, you can make it so that even if the network system does go down, your sales registers can continue to operate. By setting up one or more registers with the ability to run **Keystroke** in a "stand-alone" mode, you remove the requirement of needing the network to ring up sales.

Keystroke saves and updates its data files in "real-time". This means that when a piece of information is saved (e.g., a Sales Invoice) the program immediately updates all the relevant data files (e.g. reducing the QOH of each Inventory item on the sale). In order to do this, **Keystroke** must have complete access to all of its data files. When running **Keystroke** on multiple machines, this is done by sharing a single set of data files (on a "file server") between all the machines that will be running **Keystroke**. This type of data storage is fast, reliable, and very convenient, since everyone on the system will always be up-to-date on all information, even if it was entered only a second ago.

The disadvantage to real time data updates is that if a machine doesn't have immediate access to the shared data files, it can't update them. **Keystroke** can be run on a single machine which saves and updates the data files locally. The problem occurs when you want to combine the information entered on one machine with that entered on another. You can't simply copy one set of files onto another; information on one machine must be merged with that on another.

Keystroke comes with a utility called RecMerge, which stands for Recovery Merge (see the Recurring Invoices and Recovery Merge section in this Appendix). This utility allows you to merge in sales transactions entered on one computer with those on another. In order to be able to ring up sales when the network is down, you will need a register that can run **Keystroke** in stand-alone mode and have it set up correctly so the new sales information can be merged back into your main system by RecMerge when the network is back up and running.

For a register to run **Keystroke** in stand-alone mode, it must have its own hard disk. All other requirements are the same as running the program as a client on the network. The important part is making sure the register's hard disk has current **Keystroke** data files (so Inventory and Customer information is accurate for ringing up sales) and then adding the sales (rung up while in stand-alone mode) back into your main system.

Preparing for Stand-Alone Mode

The first step is to install **Keystroke** on each computer that will be used as a register if the network is down. You then need to make sure this machine has a current copy of the **Keystroke** data files. However, since you will be merging all sales transactions located on this machine back to the main system, you must make sure that there are no transaction files on this computer. The easiest way to do both of these things is to simply copy everything (including subdirectories) in the **Keystroke** directory of the file server to the **Keystroke** directory on the stand-alone machine. Then delete the transaction files (e.g., use the command `DEL C:\KEYSTROK\DATA\KSTRN*.DAT`, being very careful not to accidentally delete anything on the file server).

Since the data files on the stand-alone machine should be as current as possible, you should set up some type of procedure to automate the process of copying the data files on a regular basis. This can be done through a batch file, menu selection, or by an automated scheduling program. Remember that these files need to be copied *before* a disaster occurs. If your network stops working, it can be difficult (or even impossible) to retrieve the latest Inventory data from the file server hard disk.

The last step for setting up the stand-alone machine is to provide a way to quickly switch from running **Keystroke** using the network to stand-alone mode. One way to do this is to provide a "boot disk" that will bring the computer up in this mode when it is turned on. In this case, the Clerk can simply insert this disk into the floppy drive and then reboot the computer. Another way is to provide a separate menu selection to run **Keystroke** in stand-alone mode (e.g., via a batch file, Windows icon, etc.).

Once **Keystroke** is started in stand-alone mode, the program will inform the Clerk that it needs to initialize the transaction files. Although there are no further automatic prompts, the Clerk should also set the next Invoice Number and verify the Date/Time using the Number/Date function on the Transaction menu in Sales Manager. If this is not done, Invoice Numbers will probably begin with Number 1, and all new sales could be assigned an incorrect Date/Time. If running more than one machine in stand-alone mode, be sure to set the next Invoice Number differently on each machine to avoid using the same Invoice Number on different transactions (when running on the network, the program automatically takes care of this for you).

While running in this mode, Clerks must only enter new sales transactions (all older transactions will not be available until the network is restored).

Note: Only the Sales Manager module should be used in stand-alone mode. Information entered in any other module (and any changes made to database files (Inventory, Customers, etc.) will be lost.

Recovering After Using Stand-Alone Mode

Once the network is back up and running, the transaction file(s) from each machine should be copied back to the **Keystroke** directory on the file server, and then this machine should be switched back to running in normal mode (i.e., using data on the file server). Be sure to check the next Invoice Number again after switching back to the network. If more than one machine was operating in stand-alone mode, you should copy each set of transaction files to a different extension (e.g. **COPY C:\KEYSTROK\DATA\KSTRN*.DAT F:\KEYSTROK*.001** for Register Number 1, etc.).

You then need to merge each of the new transaction files into the main data set one at a time. Note that only the files with a three digit number in the file name should be merged in (e.g. KSTRN001.001). All other transaction files (KSTRN.001, KSTRNHDR.001, and KSTRNTBL.001) should be ignored.

To merge each file into the main data set, first copy it to the file name **KSTRN001.REC** (e.g. **COPY KSTRN001.001 KSTRN001.REC**). Then run **RECMRG /ADD** (Remember to use the /ADD Switch so the Invoice Numbers and dates are preserved; otherwise, the current Date/Time and the next available transaction Number will be used). You can use any other Startup Switches as needed.



Note: The above instructions are for one particular method of using the RecMerge utility. Please review the Recurring Invoices/Recovery Merge section (later on in this Appendix) for a more detailed description of the RecMerge utility, including other ways it can be used. Since systems are commonly configured slightly differently, you should also create a simple step-by-step Procedures Guide for the store manager and/or sales Clerks to follow in the event they should ever actually need to operate in stand-alone mode.


Network Version Features


The **Keystroke** Network Version is identical to the Single-User Version of **Keystroke** except for several important features. Either version will operate on a standard DOS 3.1 compatible network system; however, the Single-User Version will not allow more than one person to be actively using **Keystroke** at any one time. Therefore, the Network Version must be used whenever it is necessary for the system to be operated by more than one person at the same time.

When new transactions or database records are created using the **Keystroke** Network Version, the transaction number (e.g., Invoice Number) is not assigned until the transaction is saved or printed. This prevents transaction numbers from being duplicated, and ensures that transaction numbers are assigned in the order they are completed. In these cases, the number is displayed in parentheses to indicate that the displayed number may not be the number actually assigned to the transaction. The instant a transaction is printed or saved, the next available number is automatically assigned to it.

Keystroke is licensed on a per “concurrent” user basis. A concurrent user is defined as an instance of **Keystroke** that is running and has a Clerk logged in (whether or not the Clerk is actually doing anything). The program keeps track of the number of concurrent users and compares this with the number licensed for that serial numbered copy of **Keystroke**. If the maximum limit has been reached, **Keystroke** will not allow a new user to log in until another user logs out. This method of licensing allows you to have **Keystroke** available (and even loaded and running) on as many machines as you like since **Keystroke** will only count the number of programs actually in use.

Any machine that is currently displaying the  Change Clerk function (a box in the center of the screen with the message “Enter Clerk Number:”) is considered “locked” or “logged out” and is not counted as a current user. At this point, the program is locked and no operations can be performed until a valid Clerk is entered (“logged in”). In order to keep a machine from being counted as a current user, either exit the program or bring up the Change Clerk screen by pressing .

When logging in (i.e., after entering a new valid Clerk and Password), the program will check the number of current users. If the maximum number of users has been reached, the program will display a “maximum number of users” message, and will automatically continue once another machine becomes inactive (either by entering the Change Clerk function or exiting **Keystroke**). The user can either wait for the program to become available, or press  to go back to the Change Clerk function. The user should not just walk away from this screen because once a spot does become available, this machine will automatically continue using that Clerk Number and Security Level, leaving the machine available for anyone to walk up and use the program without requiring them to enter a Clerk Number.

***Note:** If you need to exit the program but can't because you have exceeded the number of licensed concurrent users, you can always enter in clerk number “0” (zero) and then press  on the password three times in a row. This will cause the program to terminate itself.*

NETWORK INSTALLERS: DO NOT RESTRICT ACCESS to the directory in which **Keystroke** or its data files are located. The program needs to be able to alter, create, delete, and rename files at any time; therefore, every user must have full rights to the **Keystroke** program directory and underlying subdirectories. **Keystroke** and its data files should be stored directly off the root directory of the file server in the **KEYSTROK** directory. Using a different directory will reduce the effectiveness of **Keystroke** support services. Data files should be stored in a subdirectory directly below the program directory (e.g., **KEYSTROK\DATA**).

Register Numbers

When using the Network Version, a different Register Number *can and should* be assigned to each machine. The program records the Register Number with every transaction. If a valid Register Number is not specified, the system will default to Register Number 1. The program will work correctly if more than one machine is using the same Register Number; however, if the Register Numbers are the same, there will be no way to distinguish what information was entered on which machine. The Current Register Number is displayed in the lower right corner of the **Keystroke** Logo screen, as well as towards the bottom of the Sales Manager screen.

*The Current Register Number is displayed in the lower right corner of the **Keystroke** Logo screen.*



Keystroke Logo Screen

There are two ways Register Numbers can be assigned. The most commonly used method is by using the /R Startup Switch (described in the following section) which assigns a Register Number to each machine when the program is loaded (e.g., the command: **KEYSTROK /R2** assigns Register Number 2). The other method, called “Multi-Drawer”, works by assigning a Register Number (or Drawer Number) to each Clerk. In this method, you may have more than one physical cash drawer attached to an individual machine, and each Clerk is responsible for their own drawer (“till”). If the Multi-Drawer method is used, any Register Number assigned using the /R Startup Switch will be ignored.


One of the primary purposes for the Register Number is to track sales and cash drawer amounts so they can be more easily reconciled using the Closeout function. This function tracks the starting amount in each cash drawer and then adds the payments received to determine the amount that should be in each drawer for each payment type.

Note: *The Closeout function can track the starting drawer amounts for register numbers 1 through 40. If a register number higher than 40 is used, the Starting Drawer amounts will need to be entered manually each time the Closeout function is used for that register. (All other features, including the rest of the Closeout function will work correctly).*

Multi-Drawer

As described earlier, the **Keystroke** Network Version provides the ability to have multiple workstations on a network, each with its own drawer. We call this the “Multiple Register” system and it is the most common type of configuration. The **Keystroke** Network Version also provides the ability to have multiple drawers on *each* machine. This type of configuration is called “Multi-Drawer”, and is typically used to make Clerks responsible for managing their own drawer. Each cash drawer (or “till”) is referred to as a Register Number and/or a Drawer Number, and although each workstation may be assigned a number or name, the system will no longer keep track of that information.


To use the Multi-Drawer mode, you must first mark the Enable Multi-Drawer setting in the Configuration Manager – Parameters function. Then you can configure the Open Codes, etc. for each drawer using the Setup – Cash Drawer function (also in Configuration Manager). Rather than using the /R Startup Switch (as used in Multiple Register configurations), the Register (or Drawer) Numbers in this mode are set using the Time Clock feature.

When a Clerk comes on duty and Clocks In, they will need to specify the Drawer Number for which they are responsible during the present shift. This is performed by Clocking IN using the  Time Clock function. Note that the Drawer Number field will be displayed on the Clock IN dialog box only if Multi-Drawer is enabled.



The Multi-Drawer function requires the Clerk to specify a Drawer Number when clocking in.


Operating Keystroke in Multi-Drawer mode

When using the Multi-Drawer mode, the Clerk must enter his/her Clerk Number in the  Change Clerk function before entering each transaction. This tells the system which Clerk is entering each sale and therefore also which drawer to open. The Prompt for Clerk parameter in the Sales Manager (Sales Entry – Parameters) should be marked ON to ensure that Clerks do this.

Notes: *It is possible to configure the system to have a mix of some registers with only one drawer and other registers with multiple drawers. The Enable Multi-Drawer parameter (as well as all other parameters) is stored in the Configuration data file. To accomplish this type of control, use the /C Startup Switch and turn ON/OFF the Enable Multi-Drawer parameter when necessary. If you are not actively using the Time Clock feature to track employee hours, you can semi-permanently assign a unique Drawer Number to each Clerk (no two Clerks with the same Drawer Number; e.g., make Drawer Number = Clerk Number). Once assigned using the Time Clock function, it would not be necessary to re-select a Drawer Number on each shift.*

Startup Switches

Startup Switches (sometimes referred to as Command Line Switches) are used to give special instructions to **Keystroke** (or any other program) as it is loaded. They are used to tell the program to operate in a manner different than it normally would (i.e., override the default settings).

To start **Keystroke**, you normally run the program KEYSTROK.EXE by typing **KEYSTROK**  at the DOS prompt in the **Keystroke** program directory, or by putting this command on a menu (or as an icon/shortcut in Windows).

For example, to specify Register 2, using data which is in the STORE1\ subdirectory of the KEYSTROK\ directory, and using a configuration data file that has been named KSCNF2.DAT (also placed in the KEYSTROK\STORE1\ directory), enter the following:

```
KEYSTROK /R 2 /D C:\KEYSTROK\STORE1\ /C KSCNF2.DAT
```

Note that the space between a Startup Switch and its variable is optional (i.e., **/R 2** is the same as **/R2**). The order in which Startup Switches are entered is irrelevant (i.e., **/R2 /C KSCNF2.DAT** is the same as **/C KSCNF2.DAT /R2**).

You can use any number of Startup Switches at the same time; however, please note that it may not be logical to use certain Switches at the same time (e.g., **/TUTOR** and **/D**). The “Standard Startup Switches” described below are available to most **Keystroke** programs/modules (not just KEYSTROK.EXE). Certain stand-alone utility programs that come with **Keystroke** also have available unique Startup Switches which are available only to those programs (e.g., STOREMRG.EXE). Such Startup Switches are described with the instructions for those programs.

Note: Some of the Standard Startup Switches such as **/R** are only applicable when running the Network Version of **Keystroke**.

/H, /? Help

Displays a documentation screen(s) for the program being run. This screen will contain a short description of the program as well as a list of the valid Startup Switches. These help screens are stored in the DOC\ subdirectory below the program directory. You can also review any of these screens directly by viewing and/or editing the file MODULE.DOC where MODULE is the name of the program to be run (e.g., **EDIT DOC\RPT.DOC** or **SHOWDOC DOC\KEYSTROK.DOC**).

/D Data Directory (default = DATA\)

The **/D** Switch is used to designate an alternate Data Directory. Enter the **/D** Switch in the following format:


```
KEYSTROK /D path
```

The *path* must exist and must be entered in standard DOS format (e.g., **STORE2** or **C:\KEYSTROK\STORE2**).

Keystroke stores all of the data you enter into a series of files located in the DATA\ subdirectory beneath the Program Directory (i.e., C:\KEYSTROK\DATA*.DAT). You can change where **Keystroke** will store these data files by using the /D Data Directory Switch. This Switch is useful for whenever you want to work on a different set of data, such as for practice, or for different stores. You may also need to use this Switch when the data files are being stored on a different drive than the drive from which the program is being loaded (perhaps because of the way your network is set up).

Note: You can specify any valid DOS path after the /D Switch. This includes the use of an abbreviated path instead of the full path. For example, to use the data for Store #1 which is located in the STORE1\ subdirectory below the program directory, you can enter either /D C:\KEYSTROK\STORE1\, or you can just use /D STORE1\. Both of these examples will produce the same results.

/TUTOR Tutorial Data

The /TUTOR Switch is used to run the program using the sample Tutorial data that is supplied with the **Keystroke** system. (This is the same as typing **TUTOR**  at the DOS prompt.)

The tutorial data is stored in the subdirectory TUTOR\ located below the **Keystroke** program directory. This is the same as using the /D Switch and specifying the TUTOR\ subdirectory (i.e., **KEYSTROK /D TUTOR**).

/E Data Extension (default = DAT)

The /E Switch is used to designate an alternate data File Extension. Enter the /E Switch in the following format

```
KEYSTROK /E extension
```

The *extension* must be entered in standard DOS format (i.e., up to three characters, no spaces).

Keystroke stores all the data you enter in a series of files called data files which normally have the file extension .DAT. Unless the /D Switch is also used, these will be stored in the DATA\ subdirectory. The complete name of each data file is made up of two parts: the File name (up to eight characters), followed by the File Extension, which is separated by a period (or dot; “.”). The File Extension is used to help group files and/or define the purpose of each file (Program Executables typically use the .EXE extension, Temporary files use .TMP, and Data files use .DAT). You can change the extension of the data files from DAT to something else by using the /E Startup Switch. You may want to do this to access a set of practice files or if you have a different set of data files for a different store.

Note: It is recommended that you avoid the use of this Switch and use the /D Switch instead. The /D Switch is more flexible and you will result in a much cleaner setup by using different Data Directories instead of different Extensions.

/R Register Number (default = 1)

The /R Switch is used to specify the Register Number. Enter the /R Switch in the following format:

```
KEYSTROK /R #
```

The # should be a number between 1 and 40 (e.g., **KEYSTROK /R 2**).

The Register Number is used to track on which register each transaction or payment was entered. This number is then used on reports and by the Closeout function to determine how much money should be in each drawer. If using the Network Version, a different Register Number (1-40) should be specified for each machine. If a specific Register Number is not specified, or if an invalid entry is made using the /R Switch, the Register Number will default to 1. You can use numbers between 1 and 9999, but the Closeout function only tracks starting drawer amounts for Registers 1 through 40.

This Switch only works with the Network Version of **Keystroke** (the Single-User version always uses Register Number 1). For more information, see the Network Version Features section of this appendix.

Note: *Keystroke supports the use of multiple cash drawers connected to a single computer. If you have this function enabled, the /R Register Switch will be ignored since the current Clerk (not the /R Switch) is used to determine the Register Number.*

/C Configuration (default = KSCNF.DAT)

The /C Switch is used to designate an alternate Configuration data file. The Configuration data file contains most of the information entered in the Configuration Manager's Setup and Company menus as well as in the Parameter functions found in most other **Keystroke** modules. Enter the /C Switch in the following format:

```
KEYSTROK /C filename
```

The *filename* must be a valid DOS file name and must be located in the current Data Directory. Unlike the /D Switch, a complete DOS path cannot be specified with the /C Switch.

You may need to use this Switch when working on a network with multiple machines if each machine has different equipment (or peripherals) hooked up to it. The most common use of this Switch is with a "Back Office" machine that is not equipped with a Cash Drawer when all the other "Registers" are.

The easiest way to set up a machine that requires a different configuration file is to first get the standard machines completely set up in **Keystroke**. Then create a new configuration file for the machines that don't match the standard ones. To create a new configuration data file, use the DOS COPY command to copy the default configuration file to a new file with a unique name (e.g. `COPY KSCNF.DAT KSCNF2.DAT`). Then use the /C Startup Switch to start **Keystroke** using the new configuration file (/C **KSCNF2.DAT**), and make any desired changes in the Configuration Manager and/or Parameters.

/F Form Files (default = FORMS.DAT)

The /F Switch is used to designate an alternate Form Files data file. The Form Files data file contains the information maintained by the Print Buttons function on the Tables menu in the Configuration Manager. Enter the /F Switch in the following format:

```
KEYSTROK /F filename
```

The specified *filename* must be a valid DOS file name and must be located in the current Data Directory. Unlike the /D Switch, a complete DOS path cannot be specified with the /F Switch.

Similar to the /C Configuration Switch, the Form Files Switch provides the ability to use different Form Files and/or Printers for each machine running on a network. If this Switch is not used, the program reads the Form Files data from the default data file FORMS.DAT.

The best way to utilize this Switch is to first configure the Print Buttons function on your standard machine. Then, create a second file by copying FORMS.DAT to a unique filename (e.g. **COPY FORMS.DAT FORMS02.DAT**). Then use the /F Form Files Switch to start the program using the new file (e.g. **/F FORMS02.DAT**). You can then make any necessary changes in the Print Buttons function.

/P - Printers (default = PRINTERS.DAT)

The /P Switch is used to designate an alternate Printers data file. The Printers data file contains the list of available printers and how they are configured using the Printers function on the Setup menu in Configuration Manager. Enter the /P Switch in the following format:

```
KEYSTROK /P filename
```

The specified *filename* must be a valid DOS file name and must be located in the current Data Directory. Unlike the /D Switch, a complete DOS path cannot be specified with the /P Switch.

Similar to the /F Form Files Switch, the Printers Switch provides the ability to use different printers for each machine running on a network. If this Switch is not used, the program reads the Printers from the default data file PRINTERS.DAT.

To utilize this Switch, copy the standard Printers data file to a unique filename (e.g. **COPY PRINTERS.DAT PRINT02.DAT**). Then use the Printers Switch to start the program using the new file (e.g. **/P PRINT02.DAT**). You can then go into the Configuration Manager and edit the list of printers using the Printers function on the Setup Menu.

/B Black & White

The /B Black & White Switch may be used to run the **Keystroke** program in monochrome mode on a color monitor.

Keystroke stores two sets of color settings: one for color monitors and one for monochrome (black and white) monitors. The program will automatically detect which type of monitor you are using and then use the correct color settings for it. However, some monitors (particularly black and white VGA monitors) have certain peculiarities which cause the wrong color set to be used. This results in some of the information not showing up when displayed by **Keystroke** (usually the hot keys on menus).

If you have a black and white monitor, you may need to use this Switch to tell **Keystroke** to use the correct color set.


*Note: The colors that **Keystroke** uses can also be edited in the Configuration Manager.*

/S Sound Off

The /S Sound Off Switch will disable all sound effects used in the **Keystroke** program.

/LOWMEM Low Memory, do not cache indexes

The /LOWMEM Low Memory Switch is used to turn off **Keystroke's** ability to cache (store) the indexes (sorted lists of database records) in memory. This can free up a significant amount of memory, especially if you have large database files. However, the program may run slightly slower when doing certain functions.

You can check the effect this Switch will have on available memory by pressing the  System Status key. The second line on the screen will display **HighKeyDim** followed by the number of Index Keys being cached and the amount of memory the cache is using. By using the /LOWMEM Switch, you will turn off this cache and all the memory being used by it will then be available for use in other parts of **Keystroke**.

Note: If this Switch is ON, the HighKeyDim will be 0. If HighKeyDim is not 0 and you think you used the /LOWMEM Switch, exit the program and make sure it is entered correctly.

/NOSNOW No Snow

The /NOSNOW Switch is used to help reduce “snow” or “flicker” on certain monitors (mainly older CGA monitors). There is no guarantee that this will help a flickering screen, but it may be worth a try. If it does not help, there is probably a problem with the monitor or video card.

XTKB and ATKB Keyboard Type

Older computers use a different type of keyboard than newer models. The older style is called an XT Keyboard and usually has 88 keys. The newer type is called an AT Keyboard and usually has 101+ keys. **Keystroke** will automatically try to detect which type of keyboard you are using; however, sometimes the program will detect the wrong keyboard (mainly when using a laptop/notebook computer). If some of the keys don’t work properly (typically the function keys), try using the ATKB Switch to tell **Keystroke** you are using a newer type of keyboard (the XTKB Switch is very rarely required).

/K Keystroke Resource Directory (default = Program Directory)

The /K Switch is used to designate in what directory to store the **Keystroke** Resource files. Enter the /K Switch in the following format:

```
KEYSTROK /K path
```

The *path* must exist and be entered in standard DOS format. For more information, please refer to Temporary Files in the File Management section later in this Appendix.

/U User Temporary Directory (default = Data Directory)

The /U Switch is used to designate in what directory to store the User Temporary files. Enter the /U Switch in the following format:

```
KEYSTROK /U path
```

The *path* must exist and be entered in standard DOS format. For more information, please refer to Temporary Files in the File Management section later in this Appendix.

/NOSPLIT

This Switch will disable **Keystroke’s** ability to “split” the current transaction file once it reaches its maximum size. This parameter should only be used in very special circumstances, since it can lead to loss of data if the file actually does reach its maximum size.

For more information, please refer to the Transaction Files section under File Management later on in this Appendix. You may also want to review the Split Transactions On parameter and the File Move function, both located in the Configuration Manager.

Note: The following Switches are valid only with the *KEYSTROK.EXE* command, and are not used by any other modules or programs.

/NEWFILE Split Transaction File

The /NEWFILE Switch will force **Keystroke** to immediately “split” the current transaction file and start a new one. This may be necessary when a transaction file is corrupted or otherwise needs to be preserved in its present state while still allowing new transactions to be entered.

For more information, please refer to the Transaction Files section under File Management later on in this Appendix. You may also want to review the Split Transactions On parameter and the File Move function, both located in the Configuration Manager.

/TIME Date/Time Prompt

The Date/Time Prompt Switch will cause **Keystroke** to ask for the current date and time whenever you enter the program. Since **Keystroke** is date/time oriented, this should be used on any machine that does not have a built-in clock or is experiencing problems keeping the correct time.

/G Go Direct to Module (default = Logo Screen)

The /G Switch is used to specify in which module to start **Keystroke**. Enter the /G Switch in the following format:

KEYSTROK /G module

The *module* must be a valid **Keystroke** executable file name (see list below).

When you run the **Keystroke** program, it normally begins at the Logo screen, where you are required to log in as a Clerk and select a module from the Main Menu. Using the /G – Go Direct to Module Switch, you can instruct the program to automatically start in a specific module. The /G Switch essentially bypasses the Main Menu; it cannot be used to perform any special functions, as can be done with Report Cues or Custom modules (e.g., GL-Link and Importer).

Once the specified module is loaded, the program will prompt you to specify a valid Clerk (the same as it normally does at the Logo screen). All other aspects of running the program are identical; you can then enter sales, change modules, exit the program, etc.

Executable file names for **Keystroke** modules that can be accessed using the /G Switch:

Module or Function	File Name	Custom Modules	File Name
Sales Manager	TRN	GL-Link	GLLINK
Database Manager	DBM	Script Editor	SCRPT
Purchase Manager	POR	Inventory Production	PROD
Report Manager	RPT	Importer	IMP
Label Manager	LBL	StoreLink	STR
Accounts Receivable	AR	Keystroke EDC	EDC
Configuration Manager	CNF		
Quote	QTE		
Closeout	CLOSEOUT		
Inventory Variance	VAR		

For example, to start **Keystroke** in the Sales Manager, start the program as follows:

```
KEYSTROK /G TRN
```

/NOD No Disk Space (or File Size) Check

The **/NOD** No Disk Space Check Switch instructs the program to not check the amount of available hard disk space nor the size of the data files when the **Keystroke** program is started up. This will cause the program to load slightly faster.

As **Keystroke** loads, it will normally check the amount of free space available on the disk that contains the data files (see **/D** Data Directory Switch). If there is less than one megabyte of free space available on the disk, a warning message will be displayed. Using **/NOD** turns off this check and its warning message.

Note: If the low disk space warning appears, it is extremely important to free up additional hard disk space as soon as possible. See *Managing Disk Space* later on for suggested ways to make more space available.

Keystroke will also check the size of each data file to make sure it is not reaching its maximum size (8.19Mb). If any of the files are about to reach their maximum size (i.e., if they are over 7.75Mb), a warning message will appear (see Warning #203 in the Errors section of this Appendix). You can also bypass this check by hitting any key while it is being performed while the message line says “**Checking data files...**”.

Note: Please use this Startup Switch carefully. If the **/NOD** Switch is used, there will be no warning prior to the hard disk or a file reaching its maximum storage capacity, at which point errors will occur and data will be lost.

File Management

The **Keystroke** program is made up of hundreds of individual files which work together to create the **Keystroke** system. There are several types of files and numerous files within each type. The different types include Program files, Screen and Resource files, Help files, Form files, Data files, Temporary files, Document files, and Tutorial files. Most of the files are included on the **Keystroke** installation disks (except for Data and Temporary files, which are created by the program after installation) and can be reloaded at any time by running the **INSTALL** program.

Note: For a complete detailed list of all the files included on your installation disks, and what each file is used for, view the **FILELIST.TXT** file located in the **DOC** subdirectory. You can view this file by loading it into any text editor or word processing program.

Subdirectories

When **Keystroke** is installed onto your system it is placed in a directory on your hard disk (e.g., **C:\KEYSTROK** or **F:\KEYSTROK**). This is the main **Keystroke** directory and is referred to as the “Program Directory”. If you are running the **Keystroke** program from the Windows operating environment, the Program Directory will be the same as the “Working Directory” in the Properties dialog box. **Keystroke** can only be started from this directory. All files and subdirectories are referenced from this directory.


Note: *Keystroke* can actually be installed under any directory name and on any drive letter. So it is possible (but not recommended) that your Program Directory can be different from the default `KEYSTROK\` directory.

In order to manage the hundreds of files that the **Keystroke** system uses, certain types of files are stored in special subdirectories below the Program Directory (e.g., data files are stored in `C:\KEYSTROK\DATA\`). Each of the standard subdirectories is described below. You may actually have more subdirectories on your system for different sets of data files (e.g., for multiple stores) or for on-line backup copies.

Data Directory DATA

The Data Directory is where **Keystroke** stores all data generated and used by the program. By default, the Data Directory is set to be the subdirectory `DATA\` (located below the main **Keystroke** Program Directory). An alternate Data Directory can be determined by one of two methods. First, the Startup Switch `/D` can be used to specify a different location for the data files (see the System Startup Switches section earlier in this Appendix). Or, if any data files exist in the Program Directory and the `/D` Switch is not used, **Keystroke** will use the Program Directory as the Data Directory. More specifically, if the Program Directory contains the file `KSCNF.DAT` or `KSTRN.DAT`, the Program Directory will be used as the Data Directory.

Summary Files DATA\SUMMARY

Summary files are kept in the subdirectory `SUMMARY\` located below the Data Directory. Summary files are optional and will only exist if generated by running the `SUMMARY` program described later on in this Appendix. Summary files contain the information displayed when you use the  Show Summary function. For more information on this feature, refer to the Data Files and Summarize History sections later on in this Appendix.

Old Summary Files DATA\SUMMARY\OLD

Old Summary files are kept in the subdirectory `OLD\` located below the Summary subdirectory described above. This subdirectory should only contain files moved there by using the `SUMMARY` program with the `/MOVEOLD` Switch. Data in these files only is accessible by Summary Reports and are not displayed by the Show Summary function. If you do not use these reports, files in this subdirectory can be deleted.

Forms Directory FORMS

All Form Files (`.NVC`, `.POR`, `.SMT`, `.ROA`, `.STR`, `.PTG`, `.LBL`), Printer Drivers (`.PRD`), and other miscellaneous device drivers (`.KSI`, `.KSD`, `.AUT`) are stored in the Forms Directory.

Help Directory HELP

All Help files (`.HLP`) are stored in the Help Directory. These are the files the program accesses when you press .

Tutorial Directory TUTOR

Tutorial Data files (`.DAT`) are stored in the Tutorial Directory. The program uses these files when you type **TUTOR** to start the program (instead of typing **KEYSTROK**).

Documents Directory DOC

This subdirectory contains the file FILELIST.TXT (which is a complete description of all files included on the installation disks), all short program description files (*.DOC), and header files which define fields produced by the Export function (*.HDR).

Files Directory FILES

This subdirectory contains the default startup files. Every time **Keystroke** is run, the program checks to make sure that all the required data and resource files exist. If any are missing, the program will automatically create them for you. The program will prompt you before creating any major data files (Configuration, Database or Transaction). Some of the data and resource files are created by the program while others are simply copied from a set of default files which are stored in the FILES\ subdirectory. These default files have the .SCR extension.

Installation Directory TEMP

This subdirectory is used by the installation program to store copies of the files contained on your **Keystroke** installation disks. These files are then expanded and stored in the appropriate directories. After installation is complete, all files stored in this subdirectory can be deleted without repercussion.

EDC Directory EDC

This subdirectory contains all files used by the **Keystroke** EDC Custom Module. Files in this subdirectory with the .LHT extension are the data files for this module.

Reports Directory REPORTS

This subdirectory is created for use by the **Keystroke** Report Writer and Report Generator custom modules, and is where all the customized report formats are stored. The report forms have the .RPT extension; the .CAT files are used to categorize (group) custom reports.

Temporary and Resource Files

The **Keystroke** program uses a variety of Temporary and Resource files for miscellaneous program functions. Resource files have the extension .SCR while most (but not all) of the Temporary files are logically named with the .TMP extension.

Depending on your system's configuration, it can be very important to instruct the program to locate all or some of these files in a specific directory or on an alternate disk drive. This is especially true if you are running **Keystroke** on a network. Your system's performance can be significantly improved by telling **Keystroke** to store these files (especially **Keystroke** Resource files) on a local hard disk or even a RAM disk for optimum speed.

Keystroke Resource Files /K

Keystroke Resource files (KSMENUS.SCR, KSDLG.SCR, KSSTRING.SCR) contain information that determines how most program screens will be displayed (e.g., menus, dialog boxes, and text strings). Since just about everything the program does requires the retrieval of information from at least one of these files, the program is very sensitive to the speed of accessing these files. This is especially true if running **Keystroke** on a local area network (LAN). It is therefore very important to locate these files on the fastest storage device available (e.g., a RAM disk (although not available in Windows95) or a local non-compressed hard disk).

You can specify where to locate the **Keystroke** Resource files in one of three ways. The best method is to use the Startup Switch /K. To use this Switch, add /K *path* to the command line used to start the program, where *path* represents the desired disk drive letter and/or directory where the **Keystroke** Resource files will be located (e.g., KEYSTROK /K C:\TEMP).

Another method is to set the DOS environment variable KSTEMP. This is done by adding a command line to the AUTOEXEC.BAT file on each machine (e.g., **SET KSTEMP=C:\TEMP**).

If you do not use either of these options, the program will automatically resort to using the DOS environment variable TEMP, if it exists. Since many other programs (including Windows and DOS) use this same variable for a similar purpose, your AUTOEXEC.BAT file may already include the necessary command line that defines the TEMP variable (e.g., **SET TEMP=C:\TEMP**).

Regardless of which method you use to determine where your **Keystroke** Resource files will be located, you should verify that the drive and directory exist and that you have at least 500K of available space (plus space required by other programs). Each time the **Keystroke** program is run, it will automatically copy these files from the **Keystroke** program directory to the specified disk, if necessary. The program will only copy the file(s) if they do not exist, or if they are different than the respective file in the program directory. The program also checks for available disk space, and will not copy/use this directory for *any* of the files if there is insufficient disk space. Again, these files require a total of about 500K.

Temporary User Files /U

Temporary User files are used by the program to store data on a temporary basis for the purpose of generating reports, editing Comments and Clerk Messages, printing large Forms, and various other program functions. When using the Network version of **Keystroke**, many of these files are unique to each machine or "User".

Some Temporary files are actually just quick backups of the original data file. These Temporary files will be located in the Data Directory and will have the same filename as the original data file but with the .TMP extension. Such files are created whenever the program has to make any major changes to the file (e.g., when the Pack function is used).

Other Temporary files are made by different parts of the program as a form of virtual memory. Since some functions require more memory than is normally available, the program will instead use hard disk space as defined by the /U Startup Switch.

Temporary File Names

Unless otherwise noted, the # symbol is replaced with the Machine (User) Number. The Machine Number is automatically determined by the program when it is loaded. This may (and usually will) be different each time the program is loaded. It does not necessarily represent a particular physical machine (as with the Register Number).

TEST.TMP	Used to make sure directory is valid
DataDir\CC#-P	Used by the Cash and Check Calculators in the Closeout function ("P" represents the Payment Number.)
<i>DataDir\USERS.TMP</i>	Used to keep the check-in time for each Machine Number
<i>DataDir\USER#.TMP</i>	Used to track which machines are using Keystroke
<i>DataDir\KSRPT#.TMP</i>	Used by the Report Manager
<i>DataDir\STRRPT.TMP</i>	Used by StoreLink Reports
<i>DataDir\VAR#.TMP</i>	Used for sorting items on a Variance (# = Register Number)

<code>UserDir\FI2-#.TMP</code>	Used for printing large forms
<code>UserDir\FI3-#.TMP</code>	Used for printing large forms
<code>UserDir\BFR#.TMP</code>	Used for editing Comments and Clerk Messages
<code>UserDir\TRN-#.TMP</code>	Used for updating the Back Order list
<code>UserDir\POR-#.TMP</code>	Used for updating the Open Order list
<code>UserDir\REC#.TMP</code>	Used by the Receiving function in the Purchase Manager
<code>UserDir\SS#.TMP</code>	ScreenSaver for ShowStatus function
<code>COUNT.TMP</code>	Used by the DATACAPT.EXE program
<code>EDC\SAVECARD.TMP</code>	Used by Keystroke EDC when purging old EDC Payments

The User Directory (“UserDir”) is normally set to the Data Directory. However, this can be changed by using the Startup Switch /U Temporary User files.

If implemented properly, the /U Startup Switch can cause certain program functions to run significantly faster. The ideal (fastest) configuration is to use a RAM disk which stores files in memory, which is much faster than even a local non-compressed hard disk. If your system is running on a local area network, a local hard disk should be faster than running across the network. If you are running a single stand-alone computer, this Startup Switch will only provide improvement if you can use a RAM disk, and only a slight improvement if your Data Directory is on a compressed hard disk and you use a non-compressed hard disk for Temporary User files.

To use the /U Switch to redirect Temporary User files, add /U *path* to the command line used to start the program, where *path* represents the desired disk drive letter and/or subdirectory where Temporary User files will be located (e.g., **KEYSTROK /U C:\TEMP**). The drive and directory must exist and have at least 1Mb of available space.

*Note: If using the /U and/or /K Startup Switches, make sure the drive and/or directory exists and it contains plenty of available space (minimum of 1Mb for Temporary User files, plus/or 500K for **Keystroke** Resource files). If you need assistance in setting up a RAM disk or in configuring your system to use this or any other Startup Switches, please contact your local Authorized **Keystroke** dealer.*

Note: Microsoft Windows95 does not support RAM disks.

Data Files

Keystroke data files have been structured to provide strong data integrity with maximum flexibility and speed. Different types of data are stored in separate files, but all the data files will have the same file extension (.DAT), and will be stored in the default Data Directory (DATA\). All the data files put together make up a data set. You may have several data sets available to you; each would be stored in a different directory and/or possibly have a different file extension.

All **Keystroke** systems have at least two data sets available. The primary data set is located in the DATA\ subdirectory below the program directory and uses the file name extension .DAT. This is where your data should be located. The other data set is the Tutorial data, which is supplied as a place to practice, learn, and experiment with the wide variety of functions and parameters available in **Keystroke**. The tutorial data is stored in the TUTOR\ subdirectory and also uses the file extension .DAT.

Other data sets can be created and used by utilizing the Startup Switches /D for Data Directory and/or /E for Data Extension. The two most common reasons to do this are for practice data and for multiple store applications. To set up your own set of practice data, copy your real data files into a subdirectory called PRACTICE\ and load **Keystroke** with /D PRACTICE\. If you are working with data transferred from multiple stores, load **Keystroke** with /D STORE1\, etc.

Every time **Keystroke** starts up, it will check to make sure every data file exists for the data set to be used. If any of the files are missing, the program will create them before proceeding. When the program needs to create any of the major data files (Configuration, Database, or Transaction files) it will display a message box in which you can confirm whether or not to create the new file(s). Other less significant data files will automatically be created without requiring you to confirm the action.

When the program displays the message that it is about to create a new data file, you have the option to proceed by selecting **[Ok]** or to abort by pressing **[ESC]** (or selecting **[Cancel]**). Since **Keystroke** requires all data files to be available in order for it to function properly, canceling the creation of any data file will cause the program to abort (quit).

After the program confirms that you want to create a data file, it will check to see if any startup records exist for that type of file. Such startup records are stored in the FILES\ subdirectory under the same file name as the data file but with the extension .SCR. If a startup file exists, the program will prompt you with the question, “**Initialize start up records for the ... file?**”. You then can select **[Yes]**, **[No]**, or **[Cancel]**. Selecting **[Yes]** to this prompt will copy the records from the startup file(s) to the new data file(s). This will give you a small set of generic records which will make **Keystroke** ready to use “right out of the box”. Answering **[No]** will initialize blank data file(s). Selecting **[Cancel]** will cause the program to end (same as selecting **[Cancel]** from the create data file message).

Keystroke data is stored in several different types of data files. These include Configuration/Parameter files, Database files, Transaction files, Table files, and other Miscellaneous files. The following is a description of each type of data file as well as a list of each data file use by the program.

Configuration Data Files

The primary or default Configuration Data File is KSCNF.DAT. This and other configuration files are initialized the first time **Keystroke** is operated. The Configuration Data File contains information that controls system features such as display Colors, Cash Drawers, Customer Display, Electronic Scale, GasLink, Company Name/Address information, Company Defaults, and all optional settings controlled by the Parameter functions located in the Configuration Manager and most other modules.

Separate data files are used to store configuration information entered in the Printers function on the Setup menu, Form Messages function on the Company menu, and each of the functions on the Tables menu (except Authorization Methods). Each of these data files are listed below.




Function	Data File Name	Notes
Printers	PRINTERS.DAT	Unless /P Startup Switch is used (See Printer Drivers)
Form Messages	FORMMSGS.DAT	
Security Levels	LEVELS.DAT	
Sales Payment Types	TRNPMNTS.DAT	
Purchase Payment Types	PORPMNTS.DAT	
Authorization Methods	*.AUT	Each Authorization Method is stored in a separate file with the extension .AUT located in the FORMS\ subdirectory
Print Buttons	FORMS.DAT	Unless /F Startup Switch is used
Price Tables	PRCTBL.DAT	List of Price Tables
	PRCCODE.DAT	List of Price Codes
	PRCFRM.DAT	List of Price Formulas

Tax Tables	TAXTBL.DAT	List of Tax Tables
	TAXDST.DAT	List of Tax Districts
	TAXCODE.DAT	List of Tax Codes
	TAXFRM.DAT	List of Tax Formulas
Commission Tables	COMTBL.DAT	List of Commission Tables
	COMCODE.DAT	List of Commission Codes
	COMFRM.DAT	List of Commission Formulas

***Note:** Each of the components of Price, Tax, and Commission Tables (Tables, Districts (Tax only), Codes, and Formulas) are stored in separate files. Each of these file sets should always be treated as a whole. For example, if copying Tax Tables from one data directory to another, copy all four Tax Table files (e.g., TAX*.DAT). See instructions for each of the respective functions for more information.*

Keystroke provides the ability to use an alternate Configuration Data File that can be specified when the program is loaded (/C Startup Switch). The most common reason to use an alternate Configuration File is when some machines on a network are equipped with different peripherals (e.g., some have a cash drawer while others do not). Another use is to allow different users to set up display colors to their liking. Similar Startup Switches are also available for the Printers function (/P) and the Print Buttons function (/F). To use any of these alternate files, refer to the respective Startup Switches described earlier in this Appendix.

Database Files

Database files are used to store Inventory, Customer, Vendor, Clerk, Department, and Category information. These files can be maintained either on the fly from anywhere within **Keystroke** (via  Query, or  List function with  to Edit), or by using the Database Manager module.

Keystroke database files have a maximum size of 65,535 records. This limit is because the database engine uses a 2-byte unsigned integer to keep track of the records within the file, and 65,535 (64K) is the largest number that a 2-byte integer can represent. All the database files use a 128 byte record size (which gives a maximum file size of about 8 Mb). The first 17 records of each data file are used to store header information about the file itself. This renders a maximum number of 65,518 database records that can be stored in each file.

Each **Keystroke** database is actually made up of several files. The file names all begin with “KS”, followed by the three letter file code defining which database the file applies to. For example, Inventory database files begin with KSINV, Department database files begin with KSDPT, etc. (the Category database is stored in the same files as the Department database). The rest of the filename sometimes defines what type of information is stored within the file.

The main information for a database is stored in the file without any additional characters added to its name (e.g., KSINV.DAT). This file contains all the data fields that can be indexed (for fast sorting and look up capabilities) as well as certain other fields that need to be accessed the most quickly (e.g., Quantity On Hand). The rest of the fields from a database record may be stored in additional files which will have a number at the end of the file name (e.g., KSINV1.DAT and KSINV2.DAT).

Indexes for each database are stored in the file with the characters NDX at the end of the name (e.g., KSINVNDX.DAT). The program uses the indexes to quickly locate specific records within each database file. All database Comments are stored in files with CMT, and Serial Numbers are in the file with SER (e.g., KSINVCMT.DAT and KSINVSER.DAT (Inventory only)).

The following is a complete listing of data files for each Database.

Inventory:	KSINV	KSINV1	KSINV2	KSINVCMT	KSINVNDX	KSINVSER
Customers:	KSCST	KSCST1	KSCST2	KSCSTCMT	KSCSTNDX	
Vendors:	KSVND	KSVND1	KSVND2	KSVNDCMT	KSVNDNDX	
Clerks:	KSCLK	KSCLK1		KSCLKCMT	KSCLKNDX	
Departments & Categories:	KSDPT			KSDPTCMT	KSDPTNDX	

Transaction Data Files

All **Keystroke** transaction data (nearly anything that can be sorted by date, such as Invoices and Payments) is stored within the same type of data file called a “Transaction File”. When each transaction is entered it is assigned an exact date and time. The program uses this date/time stamp to quickly look up and identify all transactions. In other words, transactions are indexed and logically organized by their date/time stamp within the Transaction Files. Throughout the rest of this section, we will use the term Date to refer to a Date/Time value.

Keystroke is very unique in how it stores many different types of transactions in a single data file (or set of data files). A single Transaction File may include all details, header information, comments and indexes for Sales (Invoices, Orders, Layaways, Quotes, and Payments), Purchases (Invoices, Orders, Reserved Orders, and Payments), ROA Payments, Variances, Time Clock entries, and Audit records (Wow!!).

In order to store so much information within a single file, **Keystroke** utilizes another unique feature of actually using multiple Transaction Files and then “chaining” them together to act as one huge file. Each Transaction File is defined by a unique Date Range and only data that fits within that range is stored within the file.

The Transaction Files are numbered and chained together to allow the program (and you) to easily switch from one file to another. The Transaction Files are named using the format KSTRN###.DAT. The first time **Keystroke** is started, it will first create a Transaction File named KSTRN001. Then, as necessary, it will create sequentially numbered Transaction Files such as KSTRN002, KSTRN003, and so on. (Transaction Files created using **Keystroke** Version 2.0 or earlier may also be named beginning with S1TRN###). The easiest way to look at this is to keep in mind that your oldest transactions are stored in the file KSTRN001.DAT, and your newest transactions will be in the file with the highest numbered file name.

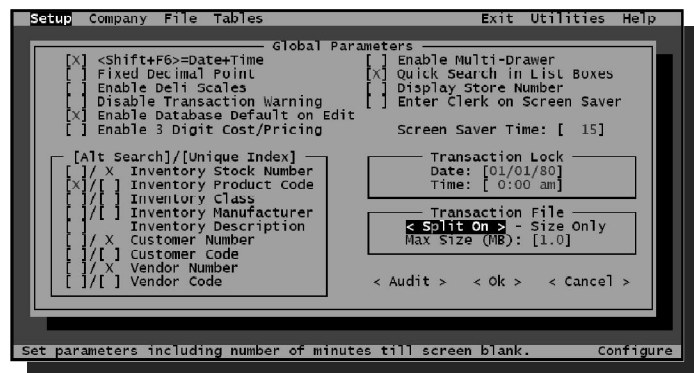
Each Transaction File is identified by a Date Range (i.e., the dates which the file encompasses). As transactions are saved, they are placed in the correct file based on each transaction’s respective Date. The highest numbered (and most recently created) Transaction File is referred to as the “Current Transaction File”. This file’s Date Range always starts with a date prior to the current date, and ends with a date out into the future (in the year 2258). Thus all new transactions that use the current date (today’s date) will be stored in this file. If a transaction’s Date is manually set to a date prior to the current Transaction File’s starting Date, it will be placed in a previous Transaction File with a corresponding Date Range. Transactions dated in the future are always stored in the Current Transaction File.

Besides the main numbered Transaction Files, there are also three other supporting files the program uses to help manage the other Transaction Files. The file KSTRN.DAT is used as a reference point to access the other Transaction Files and contains no actual data. The file KSTRNHDR.DAT contains the Next Number information to be used for each different type of transaction (e.g., the next Invoice Number), and also the Transaction Lock Date/Time. The file KSTRNTBL.DAT is used to keep track of the Transaction File names, Date Ranges, and last known location where each of the numbered Transaction Files is stored.

Note: Some of the Custom Modules (or optional programs) use this same Transaction File technology but with their own unique set of files. For example, the Production module uses files named KSPRD###.DAT and the StoreLink module uses files named KSSTR###.DAT.

Splitting Transaction Files

The **Keystroke** system continually monitors the size of the Current Transaction File. When this file reaches a certain Maximum Size (default = 1 Mb, but may be changed from 0.3 to 6.0 Mb in the Configuration Manager – Setup Parameters), the program will truncate the Current File and create a new one with the next sequential number (e.g., KSTRN002, KSTRN003, etc.).



Configuration Manager – Setup Parameters – Split On Setup

When the program truncates the Current Transaction File, it actually changes its ending date from the year 2258 to the current Date/Time (plus 10 minutes; see explanation later on under Post Dated Transactions). It then initializes the next sequential Transaction File and sets its starting date to the current Date/Time (exactly the same as the new ending date on the file being truncated). This causes the newly created Transaction File to become the “Current Transaction File”. We sometimes refer to this process as a “file split”. Note that although the previous Transaction File has been truncated, its contents are still fully available to the program. Data in it can still be edited, and data can even be added to it as long as it is dated within the Date Range of the file.

Post Dated Transactions

Every time the **Keystroke** program is started it will check the previous Transaction File for Post Dated transactions (a transaction stored within the file which has a Date that comes *after* that particular file’s ending Date). If one or more Post Dated transactions are present, a message will be displayed stating that the Transaction File should be Packed prior to running reports, editing transactions, or entering ROA payments. (You can press **F1** for Help if you do get this message and need a quick reminder of what it means and what to do about it).

Getting this message is nothing to worry about. It does *not* mean that your data is corrupted or that you are about to lose any information. However, since these “Post Dated” transactions are not located within the correct file, **Keystroke** will not be able to automatically find them, and their presence in that file can cause reports to prematurely stop before all data for the requested report has been located. This problem is easily fixed by telling the program to move these transactions to the appropriate Transaction File.

The way to move these “Post Dated” transactions is to use the Pack function (located on the File menu in the Configuration Manager). This function will automatically check each transaction in the file being packed and move it to its correct Transaction File.

There are two reasons why this problem would ever occur. The first is that a transaction’s Date was manually changed to a future Date (the program will warn you in most cases when you do this). The second cause is if you are using **Keystroke** on a network and the Date/Time is not the same on all the machines. In both scenarios, the program saves the future dated transactions in the Current Transaction File (which is where they belong for the time being), but when that file is truncated, its ending date/time is changed to the current time and any future dated (Post Dated) transactions are left behind.

Keeping Networked Clocks Synchronized

As discussed earlier, **Keystroke** will automatically stamp each transaction with the current date/time (according to the computer being used) and save it in the Current Transaction File. The problem occurs when one machine’s clock gets too far ahead of another machine’s clock. For example, if Register 1 saves an Invoice and thinks the current time is 1:30 p.m. and then Register 2 causes the current Transaction File to be split and sets its ending Date to 1:15 p.m., then the transaction from Register 1 will be in the wrong file and will be considered a “Post Dated” transaction.

In order to limit the number of times the Post Dated transaction problem occurs, **Keystroke** does buffer the file split by 10 minutes. This allows your clocks to be off from each other by as much as 10 minutes without causing a problem.

While a computer is running, it tracks time by using a system clock, which is actually a special program run internally by the computer. Since the computer is doing lots of things at the same time and since the way this clock counts the seconds as they go by is not very accurate; this system clock can lose or gain a significant amount of time each day. However, there is another clock (sometimes call the CMOS clock) within your computer that is fairly accurate (or at least more accurate). The computer looks at this clock (to set its system clock) every time you turn it on. The problem is that it only looks at this clock when the computer is turned on, and not at any other time.

If you do have a problem with your clocks getting out of sync, the best solution is to restart your computers (turn them off and then back on) at least once a day. Other solutions are to use special utility programs that will reset your system clock, either from its internal CMOS clock or from another machine on the network. Some network operating systems will automatically set your computer’s clock to match the file server’s clock each time you log on to the network.

If you get the Post Dated Transaction warning every time a file splits, it is possible you have a transaction that really is Post Dated in the future and is being pushed along each time you split a file and then pack it. If you suspect this is happening and you can’t easily locate the transaction in one of the Edit/Print lists (sort by date and press **END** to view the last transaction), you can use the Find Post Dated Transaction function in the Delete Utility (see Delete Utility later on in this appendix).

Accessing Different Transaction Files

As mentioned previously, **Keystroke** stores transactions in multiple transaction files which are chained together. In most places inside of **Keystroke**, this all happens automatically and you won't notice the program switching from one file to another. However, list boxes can only show the data from one transaction file at a time. For this reason, when viewing a list of transactions, the program will always show you the date range of the file you are viewing in the top border of the box (e.g., Transaction – Edit/Print function).

When searching for a particular transaction record in a list box (e.g., to Edit or Delete an Invoice), press **CTRL** **PG↑**, **CTRL** **PG↓**, or **CTRL** **HOME** to switch from one transaction file to another. The message bar on the last line of your screen will display these keys whenever they can be used.

If you are curious as to the actual name of the transaction file you are looking at, you can press **SHIFT** **F12** to view the System Status screen. The name of the transaction file currently being accessed will be printed in the lower left corner of the screen.

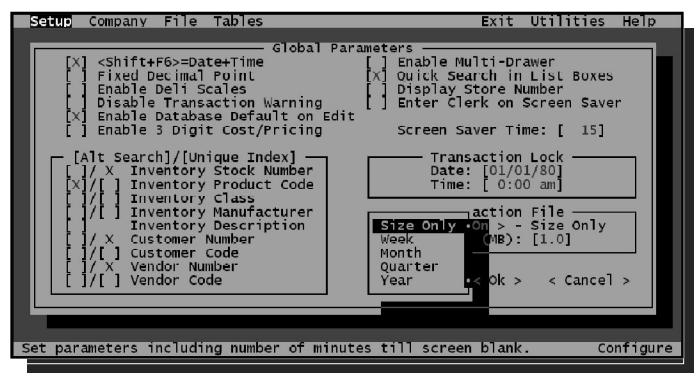
Transaction File Storage Capacity

Each 1 Mb of transaction data will store approximately 1,000 transactions (Invoices, ROA Payments, etc.). This figure may vary greatly depending on the type of transaction and volume of detail on each. This figure can be used to determine an appropriate hard disk capacity, and how often it is necessary to relocate transaction data files stored on the hard disk.

For example, if a store averages 200 transactions per day and the store is open 5 days a week, the system will probably create about 1 Mb of transaction data each week. At this rate, the system would require about 52 Mb each year for transaction data alone. This does not include Program files or Database files which can also vary greatly (see Database Files above).

Transaction files (like Database files) have an absolute maximum size of about 8 Mb each. However, **Keystroke** will split each transaction file when it reaches the predefined maximum size. You can change the Maximum Size setting for Transaction Files in the Parameters function on the Configuration Manager Setup menu.

The Transaction File Split On parameter specifies when and at what size the program will split transaction files.



File Split Parameters

As the Transaction Files split, the Transaction File Number will keep increasing from 001 up to 999. At a rate of 200 transactions per day (365 days per year, and 1000 transactions per 1 Mb file), the system can maintain complete details of every transaction for over 13 years. In other words, you will be able to pull up an Invoice from over 5 years ago and still have complete access to all of its details. If you increase your Max Size to 6 Mb, this range increases to over 80 years.

If the Transaction File Number ever exceeds 999, the program will roll over and reset to 001. Before this happens, it is important that the older Transaction Files (KSTRN001.DAT, etc.) be moved to another hard disk or subdirectory in order to make way for new transactions which will eventually be stored using the same file names. This allows the system to track an unlimited number of transactions over an unlimited range of dates.

For various reasons, it may also be convenient for some **Keystroke** users to be able to control when the Transaction Files will split in relation to the calendar. The Transaction File settings in the Setup Parameters function of the Configuration Manager allow you to do just this. You can specify whether you want Transaction Files to split based on Size Only (from 0.3 Mb to 6.0 Mb), or by periodic calendar intervals (Each Week, Month, Quarter or Year), in addition to size.

The size of Transaction Files can be set to split at anywhere from 0.3 Mb (roughly 300 transactions) to 6 Mb (roughly 6,000 transactions). The default 1 Mb file size works well for most specialty-type stores and is also very convenient, since it allows each Transaction File to easily fit on a single floppy disk. However, if you have a very high volume store (e.g., more than 1000 transactions per day), you may want to set the Maximum Size to a higher number. A good rule of thumb is to have your Transaction File size set to a number that can hold at least one week's to one month's worth of data.

Managing Transaction Data Files

Keystroke is unique in that it allows you to store and maintain access to all your transaction data in complete detail, virtually forever. However, as time goes on, the disk space required to store your sales and purchase history can become overwhelming. It may be convenient (in order to streamline daily backups) or necessary (if you run out of disk space) to periodically remove older Transaction Files from your Data Directory to a different location.

Older Transaction Files can be kept on virtually any type of device that can be used to read and write files in real-time. Tape drives do not work for this function since you cannot directly use data stored on a tape (it must first be restored). CD-ROM drives also have a limited use since you cannot edit files that are stored on them (this limitation is fine if you are just using the older files for reports and will not be making any changes to them, such as applying ROA payments).

Note: *It is not recommended to move files that will be edited unless your backup routine will include these files.*

The following is a list of places you may want to move the files to:

1. A different subdirectory on the same hard disk as the main data files.
2. A different hard disk on the same computer.
3. A different computer that is sharing its drive with the network.
4. A removable hard disk or mass storage device (e.g., Syquest or Iomega Zip/Jaz drive).
5. A floppy disk.

The above is listed in the order that will allow the fastest and easiest access to your old data. Storing to a floppy disk will only work if you have your Transaction Files split at a size that will fit on the floppy disk (this is one reason that a Max Size of 1 Mb is recommended).

When a Transaction File is moved using the Move function (accessed via the File menu in Configuration Manager), **Keystroke** keeps track of the location to which the file was moved. All **Keystroke** functions will continue to locate necessary records regardless of which data file they happen to be stored in. If the file containing a requested record can not be found, the File Path dialog box will appear stating that the program could not find a particular Transaction File and asking you to enter the "New Path" where it is located.

In the New Path field, enter the location of the file in standard DOS format. For example, if this is in a subdirectory under your Data Directory, you would enter that path (e.g., **DATA\OLDTRNS** or **F:\KEYSTROK\DATA\OLDTRNS**). If the Transaction File has been moved to a floppy disk, you would need to insert the disk containing the missing file and enter the path to the floppy drive (e.g., **A:**). **Keystroke** will then look for the file in the path you entered and if it is located, it will access the Transaction file directly from that location. If the File Path box does not go away, you've either entered the path incorrectly (double check and re-enter it), or the file is simply not located where you think it is.

The File Move function in the Configuration Manager is the best way to move your Transaction Files. This function can be used to move files out of your Data Directory, or to move them back if necessary. Please notice that this function "Moves" the file(s); it is not simply "copying". It is very important that only one copy of each Transaction File exists on your system. The Move function will copy the selected Transaction File from its source location to a destination location. It will then compare the two files to make sure that they are the same. If there were no errors, the program will delete the original file (if an error does occur, the new file will be removed). For a complete explanation of how to use the Move function, refer to the Configuration Manager section of the Main User Guide.

Note: Also refer to the Header Utility (documented in this Appendix) for more information on how the program tracks the Transaction Files and a way to repair the links if they ever get corrupted.

Database Screen Files

The Database Screen Formats for each database (Inventory, Customer, Vendor, etc.) are each stored in separate data files. The following information is presented for advanced technical reference only. Unless you are manipulating these files with the assistance of your local **Keystroke** dealer, it is really not necessary for you to study this information.

Database Screen files are named according to the following syntax: **SCRNdml.DAT**. Where *d* represents the database number (1=Inventory, 2=Customers, 3=Vendors, 4=Clerks, 5=Department, 6=Category). The *m* represents the mode of use (A=Add, E=Edit, V=View), and *ll* represents the Security Level necessary to access the file.

Keystroke ships with the following Standard screen files:

SCRN1E99.DAT	Inventory Edit mode, Security Level=99
SCRN1A99.DAT	Inventory Add mode, Security Level=99
SCRN2E99.DAT	Customer Edit mode, Security Level=99
SCRN3E99.DAT	Vendor Edit mode, Security Level=99
SCRN4E99.DAT	Clerk Edit mode, Security Level=99
SCRN5E99.DAT	Department Edit mode, Security Level=99
SCRN6E99.DAT	Category Edit mode, Security Level=99

Keystroke will automatically select which database screen format to use based on the database in use, the entry mode (Add, Edit, or View), and the Clerk's security level.

If a screen does not exist for the current entry mode, the program will use the respective screen for Edit mode. The program will use the file with the lowest Security Level that matches the Clerk's Security Level. If no matching screen file is found (i.e., the Clerk's Security Level is greater than 99), the program will use the Edit screen for Level 99. (The Variance AutoFill function and the Importer module use the Edit screen that matches the Security Level of the Clerk using the program).

These data files are edited by using the Default function in the Database Manager. This function allows you to change the default settings, links, and formulas for each field on the screen. It also allows you to assign a security level to each field. If a Clerk's Security Level is a higher number than the security level set for the field, the program will not allow the clerk to edit that field. The clerk will, however, still be able to see the information in that field. A Security Level of "0" (zero) represents "No Security Level" when used through the default screen.

Like the Screen Formats used in Edit mode versus Add mode, **Keystroke** also has the ability to automatically use an entirely different database Screen File depending on the Clerk's Security Level. For example, you could have special screens set up for "data entry Clerks" on which the on-screen position of data fields are optimized for quicker entry of just the information important to your business. Other fields could be removed so the Clerk won't need to cursor through them, and won't even see them. This capability is really designed for businesses with somewhat unusual needs; most users should be able to accomplish everything they need in this regard by changing field Security Levels through the Default function.

Note: Your local Authorized **Keystroke** Dealer can make more advanced changes to your database Screen Formats that cannot be done in the standard program. The types of changes that can be made include removing unneeded fields, repositioning fields, reducing the size of fields, and changing field titles so they can be used for different information. Changes that **cannot** be made include adding new fields and increasing the size of fields.

Summary Data Files

Keystroke has the ability to store snapshots of history information in order to provide an instant performance review for an Inventory record, Customer, Vendor, or Clerk. The files that are used to store this information are called Summary Files. Each Summary File contains the information for a single database, for a single period of time, on a single date. These files are stored in the SUMMARY\ subdirectory located below the Data directory and are named according to the following syntax:

dpymmdd.DAT Example: **ID960101.DAT**

Where "d" represents the Database (**I**nventory/**C**ustomer/**V**endor/**C**lerk), "p" represents the Period (**D**aily/**W**eekly/**M**onthly), and "ymmdd" represents the year, month, and day of the end of the respective Period.

These files are used by the  Show Summary function and by the Summary Reports. Older files are moved from the Summary Directory to the OLD\ subdirectory located below the Summary directory.

These files are like report data stored on your disk so that it can be reviewed quickly. The presence of these files will have no effect on any other part of the **Keystroke** program or your data.

Please see the Summarize History section in this Appendix for more information.

Miscellaneous Data Files

There are also a variety of other data files which are used to store miscellaneous information such as additional database information, cross referencing of transaction data, and configuration settings. These files are as follows.

ALTIC.DAT	Alternate Inventory Codes
ALTICNDX.DAT	
KSBKO.DAT	Back Order and Open Order lists
KSBKONDX.DAT	
KSCRM.DAT	Credit Memo lists
KSCRMNDX.DAT	
PRINTERS.DAT	Printer setup (also see /P Switch)
VAR#.DAT	Unposted Variance (# is replaced with the actual Register Number)
LBL.DAT	Last settings used in the Label Manager
RPT.DAT	Last settings used in the Report Manager (as well as reports that are available in other modules). RPT (# is replaced with the actual Register Number, except Register Number 1 which uses RPT.DAT)
RPT#.DAT	
RIS.DAT	Used to store data compiled by the Inventory Analysis reports
RISNDX.DAT	These files are actually temporary files and can be deleted at any time
RDS.DAT	RIS.DAT stores the totals for each item. RISNDX.DAT stores the specified sorting for the reports. RDS.DAT stores the totals and grouping information for each Department and Category
REPORTS.DAT	Stores the individual reports in the Report Cues function
RPTCUE.DAT	Stores the Cues (reports to be printed together) in the Report Cues function
REGISTER.DAT	Closeout Ending Date last used for each Register
TRN.DAT	Temporary file used by Single User version of Keystroke
SS#.DAT	Temporary files created by the Screen Blank routine when sufficient memory is not available (# is replaced with the actual Register Number) KEYSTROK.DAT Temporary file used to make sure Data Directory is valid
FORMMSG.S.DAT	List of Company Form Messages

Custom Module Data Files

The following data files are used by their respective **Keystroke** custom (optional) module. If you have not purchased the module, you will not have its data files.

GLLINK.DAT	Settings for the General Ledger Link module
CYMALINK.DAT	Settings for the Cyma Link module
KSSCR.DAT	Scripts set up using the Script Editor Module
KSSCRITM.DAT	
KSSCRCMT.DAT	
KSSCRNDX.DAT	
KSTEM.DAT	Templates set up in the Production Module
KSTEMCMT.DAT	
KSTEMNDX.DAT	
KSPRD.DAT	Production transactions created in the Production Module
KSPRD###.DAT	
KSPRDHDR.DAT	

EXTRACT.DAT	Settings to be used by the Extract function (part of MultiStore)
STOREMRG.DAT	Settings to be used by the StoreMerge function (part of MultiStore)
KSSTORES.DAT	Store setups entered in the StoreLink Module (part of MultiStore)
KSSTR.DAT	Store Transfer transactions created in the StoreLink Module
KSSTR###.DAT	
KSSTRHDR.DAT	

Managing Disk Space

As you use **Keystroke**, your data files will continue to grow, and more and more disk space will get used by the program and its data files. If you find yourself running short on disk space, there are several things you can do to increase the amount of free space available on the hard disk.

The first thing to try when cleaning up disk space used by **Keystroke** is to delete all the Temporary files. After first exiting the **Keystroke** program on all machines, you can delete all the files that have the extension .TMP from your Data Directory (e.g., type **DELETE F:\KEYSTROK\DATA*.TMP**). Be aware that **Keystroke** will automatically recreate these files as needed and you may find that the disk space freed by this step will be quickly used up again.

Depending on the volume of transactions your business enters into **Keystroke**, you may want to move your older transaction files out of the Data Directory. For more information on this topic, see the Transaction Data Files section in this Appendix, as well as the Move function in the Configuration Manager.

If cleaning up your **Keystroke** files does not free up enough disk space, you should consider removing other programs (and/or corresponding data) that are not being used. The last option is to add another hard disk to your system. This can be a very economical and cost-effective solution. Hard disks are relatively inexpensive, and adding another one to most systems is a fairly simple process.

***Note:** The Pack function does recover disk space for records that have been deleted from your data files; however, unless you've deleted a significant number of database records, this will not yield much free disk space. The Pack function creates a temporary backup of the file being Packed by renaming the original file to the .TMP extension. If you have used the Pack function, deleting the .TMP files can free up quite a bit of disk space for future use.*

Printer Drivers

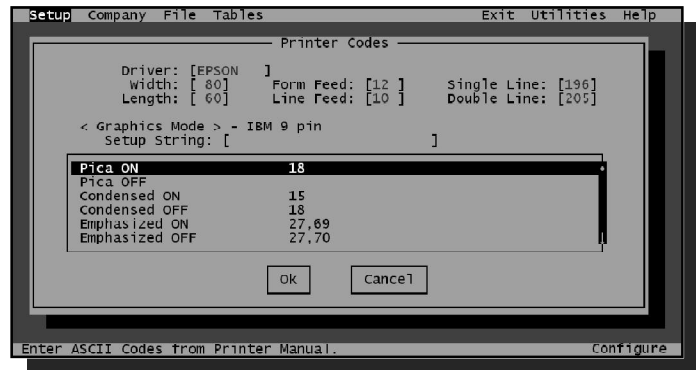
A Printer Driver is a file which contains special Control Codes (also sometimes called “escape sequences”) that can be used to instruct a printer how to print or perform other functions. Because nearly every printer has different capabilities (e.g., the ability to print different fonts or to cut the paper after a receipt), every printer may require a different Printer Driver. Sometimes, even if two printer models have the same capabilities, they may require different Printer Drivers since they are manufactured to interpret different Control Codes for the same function. This holds true even when both printers are said to be “Epson compatible”.

Unlike software programs written specifically for the Microsoft Windows environment, most text based (or DOS based) programs (including **Keystroke**) do not use the Printer Drivers supplied for Windows. Instead, **Keystroke** uses its own Printer Drivers which can be created or edited as needed from within **Keystroke**. However, since most of the information entered in the Drivers is actually only used in special situations, it is somewhat unlikely that one of the standard Printer Drivers supplied with **Keystroke** would not be sufficient to get proper printer output.

Most standard reports, forms, and labels printed by **Keystroke** use very little (if any) information from the Printer Driver. The Control Codes that are used by **Keystroke** are very similar for most printer models. If your printer is not printing at all or is printing in a manner that is too slow or not readable, the problem most likely lies either in the printer, the cable, the port, or the way it is set up, and is *not* a problem with the **Keystroke** Printer Driver. For instance, the only fields actually used by the standard reports are: Width, Length, Form Feed, Single Line, Double Line, Setup String, and Condensed ON/OFF.

Printer drivers can be selected, added, and/or edited from the Printers function on the Setup menu in the Configuration Manager. After selecting a numbered printer from the list box (e.g., **1 Report**), you can move the cursor to the list box of Drivers and either press **ENTER** to select one, **F3** to edit or view a Driver file, or press **INS** to add a new one.

Note: If you edit one of the standard **Keystroke** Drivers, be sure to give it a different file name when you save it. Otherwise, any update to the **Keystroke** program will replace the changes you made.



Editing Printer Codes for a Printer Driver

The Printer Codes box is used to specify the codes that tell the printer how data should be printed. If the wrong Printer Driver is used, or if incorrect codes are entered, the printer may print “junk” rather than the information that is desired. The codes that should be entered on the Printer Definition box are ASCII print codes, (in decimal form), that are found in most printer manuals. (Codes may also be entered in Hex format by pressing the **F9** key first). The following is a description of each of the driver fields. The most common value for each of these fields is shown in parentheses following the description.

- Driver** unique file name which is used to store the Printer Driver information. When editing a Driver, changing this field name will cause a new Driver file to be added to your system
- Width** number of columns the printer is capable of printing across a single page using its standard print size (typically 80)
- Length** number of lines per page. This number should always be greater than 20. While it may be less than the actual number of lines supported by the printer, this number should never be more than supported by the printer (typically 60)

- Form Feed** decimal form of ASCII code to cause a formfeed (typically 12)
Line Feed decimal form of ASCII code to cause a linefeed (typically 10)
Single Line decimal form of ASCII code to print a single horizontal line (typically 196)
Double Line decimal form of ASCII code to print a double horizontal line (typically 205)

Note: If you get slanted characters (usually D's and M's) instead of single and double horizontal lines, either change the symbol or character set on your printer to the "Line Draw" or "Graphics" set. If you can't do this, change the ASCII numbers to **45** and **61**.

- Graphics** determines the graphics mode for printing barcode labels. Select this button to bring up a list of the available modes. This setting is only important if you are printing barcode labels
- Setup String** print codes sent to the printer at the beginning of each print job. This can be used to set margins or symbol/character sets
- Various Character Type Codes** The remaining fields on the Printer Codes dialog box are used for the various styles of printing and optional ON/OFF Codes. To edit the control codes to be sent for each print style, highlight it, press the **ENTER** key, and enter the appropriate code for your printer. The codes are entered either in ASCII decimal (code 65 represents a capital A) and/or character format (just enter A). Any numbers are interpreted as ASCII Decimal Codes (multiple codes on the same line should be separated by a comma). You can switch the code format between Decimal and Hex mode by pressing **F9**. Each character style has an ON and OFF code. ON codes are always sent before a field is printed and OFF codes are sent after a field is printed. If your printer is not capable of handling a particular type style, either leave that field blank or enter codes for a similar style

Note: The names of the Control Codes are stored in the file `CODENAME.TXT`. This file can be modified using a standard text file editor to change the appropriate titles. This file is shared by all Printer Drivers and formatting functions, so any changes you make will be reflected on all drivers and screens that display the Printer Code list. All the functions that use Printer Codes actually refer to them by number (in order of how they appear on the list in the Printer Driver). Changing the names of these codes has no effect on how they are actually used (e.g., changing title of the Bold code to Italic will not cause fields marked to be printed Bold to start printing Italic. However, entering the Italic control code for your printer in the Bold field will cause everything that's supposed to be bold to then print in italics).

Customer Display Control Files

Keystroke can cause a wide variety of information and messages to be displayed on a Customer Display unit.

The Customer Display Control File defines what information is passed to the optional Customer Display unit. There are 18 different messages that the program can send during various program operations. The Control File determines exactly what is sent at each stage in the program.

To edit or create a Customer Display Control file, select the Customer Display function from the Setup menu in the Configuration Manager module and select the **<Display>** button. From the Display File list box, press **F3** to edit an existing file, or **INS** to create a new file. It is much easier to edit a current file and change its name (thus creating a new Control file) than it is to start from scratch.

The Customer Display Control File dialog box consists of three columns: the Message Line Description (left column), the Pause Time in seconds (center column), and the Control Sequence Data that defines what information will be sent to the Customer Display (right column).

The Message Line Descriptions are fairly self explanatory. If there is any question, a test message can be entered to determine when a message is displayed.

The Pause column in the center is used to define the minimum number of seconds that the message will be displayed. The program uses a queue to send the messages to the Customer Display unit. This allows the program to continue to run even if the pole display is backed up due to the pause times entered for what is being displayed. For example, if you enter a 3 in the pause column of the Line Item display line, the program will always display the information for a new line item entered on a sale for at least three seconds. If you are entering an item every two seconds, the pole display will not necessarily be displaying the same item as the one being entered due to this delay.

The main purpose of the pause option is for displaying the Sales Tax message (or amount). Since the Sales Tax message is immediately followed by the Total message (without a pause), it would otherwise not be displayed long enough for the Customer to see it. Therefore, you may want to induce a short pause on the Sales Tax line. All other message lines generally do not require a pause and can be left at zero since there is normally a natural pause between data fields.

In the larger column on the right, enter the appropriate control sequence data that is to be sent to the display unit. Each line may contain up to 100 characters. The program will send exactly what is entered in this field except for the following special types of entries:


- {#}** ASCII Decimal Codes: Numbers inside of squiggly braces will cause their corresponding ASCII character to be sent (e.g., {13} will send a Carriage Return). This type of entry should be used to send special control codes as defined by the display unit
- [#, #, J]** Variable Data Fields: This format in square brackets is used to send data which changes as the program is used (such as totals and descriptions). Inside the brackets enter the Field ID, the number of character spaces to print it in, and the justification (Left, Right, or Center) to be used. For example: [5, 9, R] will send the line item Price, using 9 characters, in Right justified format)

Use the Customer Display Control File to set up a wide variety of messages which will be displayed during a sale.



Customer Display Control File

Field ID

The following is a list of Field ID variables for the Line Item, Payment, and Other Transaction messages. This list is also available from the  Help Screen when editing a Customer Display Control File.

Line Item Message:

- 1 Stock Number
- 2 Product Code
- 3 Description
- 4 Class
- 5 Price
- 6 Quantity
- 7 Extended Amount
- 8 Tax Code
- 9 Department
- 10 Category
- 11 Manufacturer
- 12 Long Description
- 13 Base Price Discount
- 14 List Price Discount

Payment Message:



- 1 Payment Name
- 2 Amount
- 3 Reference
- 4 Food Stamp Total
- 5 Food Stamp Remaining

Other Transaction Messages (Totals):

- 1 Customer Number
- 2 Customer Name
- 3 SubTotal
- 4 Discount
- 5 Shipping
- 6 Tax Table
- 7 Sales Tax Rate
- 8 Sales Tax
- 9 Total
- 10 Change Due
- 11 Amount Due
- 12 Customer s Amount Due
- 13 Customer s Available Credit
- 14 Food Stamp Total
- 15 Food Stamp Remaining
- 16 Base Price Discount
- 17 List Price Discount
- 18 Customer Code

Display Strings

The following describes when each display string is sent to the Customer Display. Note that Variable Data Fields are not available for the Setup, Inside **Keystroke**, Leaving **Keystroke**, Enter Clerk (Locked), and Waiting for New Sale messages.

- Setup** when first entering **Keystroke**, or whenever information is changed in the Configuration Manager module
- Inside Keystroke** when changing to a new module
- Leaving Keystroke** when exiting the program
- Enter Clerk (Locked)** when prompted to enter a Clerk Number (access to program functions are disabled until a Clerk and Password are specified)
- Attached Customer** when a Customer has been specified on a transaction
- Line Item** when a line item has been entered on a transaction
- SubTotal** when the Complete Sale box is displayed
- Discount** when leaving the Discount field
- Shipping** when leaving the Shipping field
- Sales Tax** when a new payment is being selected
- Total** when a new payment is being selected (follows the Sales Tax display)
- Payment Entered** after a new payment has been entered
- Change** when on the Print box and the Amount Due is \$0.00
- Amount Due** when on the Print box and the Amount Due is not \$0.00
- Waiting for New Sale** before any information has been entered on a new transaction
- Food Stamp Payment** after a new Food Stamp payment has been entered
- Line Item Changed** after information on a previous line item has been edited
- Line Item Deleted** whenever a line item is removed by pressing  

Label Form File Structure

The Label Form Files used by **Keystroke** are actually ASCII text files and can be edited using any standard text editor (as well as by the **Keystroke** Label Editor). In order for **Keystroke** to use a label form file, it must be stored in the FORMS\ subdirectory below the Program Directory. **Keystroke** supports two types of Label Form Files: Price Tags (which have the .PTG file extension) and Mailing/Shipping labels (which have the .LBL file extension). Both of these types of files have the exact same format. The only difference is in the Field ID list that is used by each type (see below).

Notes: Extra blank lines in a Form File will cause an "Error 62" when **Keystroke** reads the file. Some text editor software programs automatically add an extra line to the bottom of files, thus causing a system error. If this happens, simply remove all extra lines from the Form File using a different editor program.

Form Files must be written in the following structure (the **Keystroke** Label Editor does this for you). The first line of every Form File represents the Header and each line thereafter determines an individual Field description. The Header defines the overall layout of the Form File.

Header: "Description", Control Flag, Number of Lines,
 " ", Number Across, Label Width

Field: "Text", Field ID, Line Number, "Alignment
 Code", Field Width, Print Code

Header Parameters

Description

The Description Field is not actually used by the program. It is available for you to make a note as to what the label does.

Control Flag

The Control Flag is used to turn ON and OFF the Skip Blank option. It should be set at "0" for Price Tags, and "1" for Mailing and Shipping labels. Setting the Control Flag in the Header to "1" causes fields containing no data to be suppressed on the printed label. For example, if any part of a Customer record is not completed, (such as Business Name or Salutation) a Control Flag of "1" in the Header will prevent the labels from having gaps or blank lines.

Number of Lines

The Number of Lines field represents the vertical height of each label. If a label is one inch high, the Number of Lines would be 6 (based on a standard six lines per inch).

Number Across

The Number Across field represents the number of labels to be printed side by side.

Label Width

The Label Width field represents the horizontal width of each label. This is specified in terms of Pica characters, or tenths of an inch. For example, if a label is one inch wide, the Label Width would be 10.

Field Parameters

Text

The Text field can be used to enter normal text (or space characters) to be printed on every label. All text must be enclosed in quotation marks, and the Field ID must be "0".

Field ID

The Field ID defines what **Keystroke** variable field will be printed. Below is a list of all the available field I.D.s for Price Tags, Customer Mailing/Shipping Labels, and Vendor Mailing/Shipping Labels:

Price Tag Field ID s			
Field Description	ID	Field Description	ID
Text	0	Current Date	19
Stock Number	1	CurrentTime	20
Product Code	2	Vendor1-PartNum	21
Description	3	Inventory Comment	22
Current Price	4	Vendor1 Vendor #	23
Sale Price	5	Vendor1 Cost	24
Normal Price	6	Price Code	25
List Price	7	Tax Code	26
Average Cost	8	Serial Number	27
Taxable Status (+ tax on taxable items)	9	Sale Unit	28
Department	10	Order Unit	29
Category	11	Manufacturer	30
Class	12	Extended Description	31
Last Cost	13	Commission Code	32
* PO Number	14	Price Level 1	33
* PO Date	15	Price Level 2	34
* PO Vendor #	16	Price Level 3	35
* PO Vendor Name	17		
* PO Invoice No.	18		

** available only when printing
By Purchase.*

Customer Label Field ID s		Vendor Label (*See Note Below)	
Field Description	ID	Field Descriptions	
Text	0	Text	
Company Name	1	Vendor Name	
Individual Name (Salut + 1stName + LastName)	2	Contact	
Billing Address	3	Billing Address	
Billing City St, Zip	4	Billing City St, Zip	
Shipping Address	5	Shipping Address	
Shipping City St, Zip	6	Shipping City St, Zip	
Customer Number	7	Vendor Number	
Alt Name/Address	8	Alt. Name/Address	
Last Name	9	<i>n/a</i>	
First Name	10	<i>n/a</i>	
Phone	11	Phone	
Code	12	Code	
Tax Table	13	Tax Table	
Price Table	14	<i>n/a</i>	
Discount %	15	<i>n/a</i>	
Amount Due	16	<i>n/a</i>	
Billing City	17	Billing City	
Billing State	18	Billing State	
Billing Zip	19	Billing Zip	
Phone 2	20	Phone 2	
Credit Limit	21	Credit Limit	
Last Purchase Date	22	Last Purchase Date	
Total Purchased	23	Total Purchased	
Salutation	24	<i>n/a</i>	
Shipping City	25	Shipping City	

Customer Label Field ID s		Vendor Label (*See Note Below)
Field Description	ID	Field Descriptions
Shipping State	26	Shipping State
Shipping Zip	27	Shipping Zip
Tax Number	28	Tax Number
Last Payment Date	29	n/a
Grace Period	30	Terms
Title	31	Customer Number (as defined by your Vendors)
Customer Comment	32	Vendor Comment
BirthDate	33	n/a
ExpDate	34	n/a

Note: The Form Files for Mailing/Shipping Labels are interchangeable for Customers and Vendors. Field Names for Vendor Labels are listed above, since several are used for slightly different purposes on Customer Database records versus Vendor records.

Line Number

The Line Number controls the vertical position of each Field on a price tag or mailing label. The Line Number of each Field must be an integer between one and the Number of Lines on the label.

Alignment Code

The Alignment Code primarily controls the horizontal position of the data or text within the Field Width. (See the Label Editor section for a list of alignment codes).

Print Code

The Print Code controls the print style of each Field on a price tag or mailing label. To combine more than one print style, simply add the Print Codes and enter the sum. For example, to print characters both Condensed and Underlined, enter a Print Code of **3084**, (12 + 3072). Problems that are often encountered when dealing with Print Codes are usually due to the fact that not all printers are capable of printing certain print styles, and the Print Codes that are used to produce each print style vary with different printers. The best way to avoid such problems is to check the printer capabilities and Print Codes in the Printer Driver Configuration.

Print Style	Code
Pica	3
Condensed	12
Emphasized	48
Enhanced	192
Bold	768
Underline	3072
Italicized	12288
DoubleWidth	49152
DoubleHeight	196608

Each code as shown above is actually a combination of the ON and OFF code for each style. You may enter just the ON or OFF portion if you wish. The ON code can be found by dividing the above code by three (**Code / 3**) and the OFF code can be found by multiplying by two then dividing by three (**Code * 2 / 3**).

Sample Labels

Price Tags

SMALL4.PTG

"" ,0,6,"",4,0
"" ,0,2,"L",1,1
"" ,1,2,"L",9,1
"" ,0,3,"L",1,4
"" ,3,3,"L",15,4
"" ,0,3,"L",1,4
"" ,0,4,"L",1,4
"" ,0,4,"W",15,4
"" ,0,4,"L",1,4
"" ,4,5,"R",10,769

LARGE.PTG

"" ,0,6,"",1,0
"" ,1,1,"L",5,1
"" ,2,2,"L",20,1
"" ,3,3,"L",30,1
"" ,4,5,"R",30,196656

BARCODE.PTG

"" ,0,6,"",1,36
"" ,1,1,"B5",12,13
"" ,4,3,"R",30,768
"" ,1,4,"L",10,3
"" ,2,4,"R",20,3
"" ,3,5,"L",30,3

Mailing/Shipping Labels

MAIL1.LBL

"" ,-1,6,"",1,0
" " ,0,2,"L",2,1
"" ,1,2,"L",33,1
" " ,0,3,"L",2,1
"" ,2,3,"L",33,1
" " ,0,4,"L",2,1
"" ,3,4,"L",33,1
" " ,0,5,"L",2,1
"" ,4,5,"L",33,1

SHIP1.LBL

"" ,-1,6,"",1,0
" " ,0,1,"L",2,1
"" ,7,1,"L",5,1
" " ,0,2,"L",2,1
"" ,8,2,"L",33,1
" " ,0,3,"L",2,1
"" ,2,3,"L",33,1
" " ,0,4,"L",2,1
"" ,5,4,"L",33,1
" " ,0,5,"L",2,1
"" ,6,5,"L",33,1

Other Form Files

The print formats of forms such as Invoices, Receipts, Quotes and Purchase Orders are also customizable. However, because these print formats are considerably more complicated than those for price tags and mailing labels, this can only be done by **SBS**, or an Authorized **Keystroke** dealer.

Using Keystroke Data With Other Programs

While **Keystroke** provides you with a very powerful and flexible system, no program can do everything for everyone. Some of the most advanced features of **Keystroke** include its ability to work with other programs to provide a complete solution for each individual store. Through the functions and utilities described below, it is possible to completely automate the extraction and updating of information stored by the **Keystroke** program.

Most of these utilities can be operated from a DOS prompt. This means that any program that can launch a DOS program can control the extraction and importing of data both in and out of **Keystroke**. Examples of programs with this capability include: DOS Batch Files, Remote Access Scripts (e.g., Closeup and PCAnywhere), other DOS programs (e.g., Lotus 1-2-3, dBase, Telemagic, etc.), customized DOS programs written in a programming language (e.g., Quick Basic, C, Pascal, dBase, Clarion, etc.), Windows programs (Word, Excel, Fox Pro, Access, etc.), and customized Windows programs written in a programming language (e.g., Visual Basic, Delphi, Visual C++, etc.). All of these programs have the ability to run several different commands in order to extract various information from **Keystroke**. They can then take this information and collect it, combine it, and re-display it in ways that fit their needs in order to help you run a more profitable business.

ASCII Delimited Data Files

All **Keystroke** data files are stored in a proprietary binary data format that is not easily understood by other programs. However, your data belongs to you and you should always be able to get to it and do whatever you want to with it. For this reason, **Keystroke** has the ability to save any of its data into an ASCII delimited file that can be understood by other programs.

The ASCII delimited file format is the most widely used way of transferring data from one program to another. Some programs refer to this format as: quote/comma delimited, comma separated variables (.CSV), or MS-DOS text (.TXT). Whatever the terms, almost all programs with import capabilities can work with data from this format. Exceptions to this rule are programs (such as Quicken) that can only import data stored in their own proprietary format. The ASCII delimited format used by **Keystroke** separates each variable by a comma, encloses string (non-numeric) variables in quotes, and separates each record with a carriage return/line feed combination (CrLf). You can generate these files either through the Export function in the Configuration Manager, or by printing a report to file with the Delimited setting turned ON.

Extracting Data From Keystroke

Data can be copied from **Keystroke** to an ASCII Delimited file using the Export function in the Configuration Manager, or nearly any report function in the Report Manager.

While the Export function does provide complete access to the data stored by **Keystroke**, it gives it to you in a raw form and possibly provides a lot more information (fields) than you really need. Sometimes a better way to get the data you want is to print a report to an ASCII delimited file (instead of sending it to a Printer). This can be done on most reports by selecting <Print To> – File and then entering the <File Name> and turning ON the Delimited setting.

Another method of extracting data from **Keystroke** is to trap the data as it is created. **Keystroke** is capable of printing any of its forms (Receipts, Invoices, Quotes, etc.) to a file instead of a printer. Once printed to a file, another program can take that file and save the new information or reformat it and print out a very customized form such as a ski lift ticket (see the FORM2LBL.EXE utility later on).

Reports can also be sent to a file for the purpose of “dressing up” with a word processing program. To do this, leave the Delimited setting OFF and open the resulting file in your word processor.

Note: If importing into a program that supports the ASCII tab delimited file format, use that format since it is usually much faster and more reliable than quote/comma delimited.

Updating Keystroke Data Using Other Programs

Data can not only be extracted from **Keystroke**, but you can also update some of the data in **Keystroke** by using other programs. This is done through the use of Import features.

Keystroke's Import module (an optional add-on program) can be used to update Inventory, Customer, and Vendor information. This includes adding new records (e.g., transferring a Customer from a prospect list to your **Keystroke** Customer database), as well as updating existing ones (e.g., resetting Inventory order points based on seasonal calculations from a spreadsheet program). It is also common for Vendors to provide their Inventory data (including costs, etc.) on disk, which can be imported as new records and/or to update existing records. Please see the Importer module documentation in the Custom Modules Appendix for more information.

Keystroke also supports the Importing of new transactions (both sales and purchasing) through the Import function found on the Special menu in the Purchase and Sales Manager modules. This function gives you the ability to create a transaction in another program and then transfer it into **Keystroke** to finalize it. One application for this is through the use of a hand-held data capture device. A salesperson can walk through your store with a Customer, scanning items as the Customer selects them. Once the Customer is through selecting items, the hand-held device can then transfer the new sale into **Keystroke**, where payment information is entered and packing slips and Invoices are printed.

Direct Access from DOS Prompt

As mentioned previously, **Keystroke** provides you with the tools needed to expand the capabilities of the system by transferring data between **Keystroke** and other programs. The best way to do this is by automating the whole process with another program and instructing **Keystroke** through the DOS interface to provide the data required.

Every **Keystroke** utility which has a DOS interface also has built in documentation on how that interface works. This documentation can be viewed by running the program with the /? Switch or by editing the file having the same name as the executable program file (.EXE), except with the extension .DOC.

The functions that can transfer data into and out of **Keystroke** from the DOS prompt include: printing Report Cues, Exporting data, the General Ledger Link, the Importer module, and importing Alternate Inventory Codes and AR balances. **Keystroke** also allows you to automate the running of procedures that manipulate the data within **Keystroke** (but do not transfer data with other programs). These functions include: Automatic Productions, Inventory Update, and Summarize History. All of these procedures are described elsewhere in this User Guide.

***Note:** The ability to access **Keystroke** modules and utilities directly from the DOS prompt are relatively advanced features and require some knowledge of DOS commands. Although the necessary commands are described fully herein, the actual incorporation of these commands into automated procedures (e.g., polling scripts or batch files) is beyond the realm of this documentation. If you need further assistance, please contact your local **Keystroke** dealer.*

Pay Off AR Balances ARPAYOFF.EXE

This function will pay off all open items (except Credit Memos) on all Customers. This is the same as the Automatic Payments function in the AR module with Pay Off set to All and no filter set. Please see the Accounts Receivable module section in the main User Guide for more information on the Automatic Payments function.

This function supports the standard Keystroke command line switches. It also supports the following special switches:

/NOP

No Pause – eliminates warning prompt.

PriorTo=

Only items dated before this date will be paid off. The default is the current date (e.g., PriorTo=10/14/97).

Ref=

Indicates text to be entered in ROA payments field; the default is blank. Information to be entered must be enclosed in quotes (e.g., Ref="Automatic Pay Off Payment").

PayNum=

The payment type to be used to create each ROA payment; the default is the payment specified in Accounts Receivable parameters. This is the Payment Number as seen in parentheses in the list of Sales Payment Types in the Configuration Manager, or as shown on the Payment Types report.

Report Cues RPT.EXE

The Report Cues function is a powerful feature that allows you to pre-define report settings and groups of reports ("Cues"). Report Cues can be run directly from the DOS prompt (without user intervention), making them easy to incorporate into automated procedures.

Setting Up Report Cues for DOS Commands

To run Report Cues from the DOS prompt, you must first set up your Report Cues in the Report Manager module. A complete description of how to set up Report Cues is included in the Report Manager section of this User Guide.

When setting up the Report Definitions for Cues, keep in mind that the main purpose of using Cues in this manner is *automation*. Therefore, Report Cues that are to be run directly from DOS should only include Report Definitions that do not prompt for additional information such as individual items or Departments.

For the same reason, be sure all Reports are set up to print to either a valid Printer or File. If Reports are being sent to a File, be aware that any files that already exist with the same File Names will be overwritten. If this is a problem, be sure to include commands in your batch file (or other program) to copy or rename the original files before running reports.

To verify that your Report Cues are configured properly for full automation, you can easily test them by running each one from the Report Manager. The program should not prompt for anything except perhaps the Date/Time Range, which can be automated using the Switches described later on. If a Report Cue is not properly configured for automation, it can still be run using DOS commands; however, it will halt when additional information or confirmation is required.

When Report Cues are run from the DOS command prompt, Security Levels are essentially bypassed since the program automatically assumes a user Security Level of zero. When the Report Cue is complete, it will return to the DOS command prompt (or the program from which it was launched).

DOS Commands for Report Cues

To run a Report Cue from the DOS command prompt, use the following syntax:

```
RPT CUE="CueName" START=date END=date [OtherSwitches]
```

The Cue Name must be enclosed in double quotes and entered exactly as it is displayed in the Cues function of the Report Manager. The START and END Switches are optional, but are usually necessary on any type of transaction- or date-sensitive reports. Other Switches may be any of the standard Startup Switches described earlier in this Appendix. The program will assume default settings (described later on) for all other Switches that are not specified. For example, if the Start and End Switches are not specified, the Date Range for the report will be for the current day only.

Using Optional Switches

The Cue Name is the only Startup Switch that is always required; however, you may need to specify other Startup Switches to generate the reports you need. Specifically, any date-sensitive reports may require the Start and End Switches.

Start={Date, M}

The Start Switch is used to specify the Starting Date on transaction related reports. If a Start Date is not specified, the program will assume the default Start Date of the current date at 0:00 am (i.e., Midnight this morning).

A specific Date can be specified using any date entry format that is acceptable to the **Keystroke** program (e.g., 1/1/97, 010197, 1.1, 1, -1d, -2W, -1Y, etc.). If not specified, the current Year and/or Month are assumed (e.g., if current system date is January 1997, **START=1** will represent 1/1/97).

You can also use the special Midnight option, **START=M**. This option will allow the report to be run for either today or yesterday, depending on when the report is started. If the report is run before noon, the Start date will be Midnight yesterday morning, so yesterday's activity will be included. If it is run after noon, the Start date will be Midnight this morning (today's activity only).

End={Date, M}

The End Switch is used to specify the Ending Date on transaction related reports. If an End Date is not specified, the program will assume the default End Date of the current date at 11:59 p.m. (i.e., Midnight tonight).

A specific Date can be specified (the same as with Start Date), or you can use the special Midnight option, **END=M**. The Midnight option represents the nearest Midnight to the current time. For example, using the Midnight option, if the Report Cue command is run at 10:00 p.m., the End Date would be the next Midnight (2 hours later). If the same Report Cue command is run at 2:00 am, the End Date would be the previous Midnight (2 hours earlier).

Other Switches

The Report Cue command may also include any other applicable Startup Switches that are available to the **Keystroke** program (described earlier in this appendix). The most typical Switches you may need for Report Cues are /D, to specify an alternate Data directory and /LOWMEM to disable the caching of indexes (which allows the program to load faster; most reports don't use the cache anyway).

Export from DOS Prompt KSEXPORT.EXE

The Export function (File menu in Configuration Manager) can be run directly from the DOS prompt by using the stand-alone utility program KSEXPORT.EXE. This program can be used to copy information in complete detail from the **Keystroke** data files to ASCII delimited files. A list of the data fields and what type of data they may contain is located in the file EXPORT.DOC, located in the DOC\ subdirectory. You can view this file using any text editor.

Using the DOS Export Utility

Using a variety of Startup Switches, all options available in the standard Export function (located in the Configuration Manager) can also be utilized with the stand-alone Export utility program. This utility also supports the same Startup Switches available for the **Keystroke** program (e.g., /D, /LOWMEM, etc.).

To run the Export utility from the DOS command prompt, use the following syntax.

```
KSEXPORT TYPE=I SORT=1 FILE=EXPORT.TXT FILE2=DETAIL.TXT
START=date END=date FILTER=1"condition" TAB=OFF
[OtherSwitches]
```

Default settings are used for any Switches that are not specified. Most default settings are exactly the same as displayed in the standard Export function in the Configuration Manager. The entire command line must be entered on one line (syntax above is displayed on multiple lines due to space limitation only).

Complete data field listings (including field names and order) are available for each Type of Exportable data in the DOC\ subdirectory, and all have the .HDR extension. Some programs can read these .HDR (called Header files) to provide a list of the fields being imported (or “merged”).

Type={S, A, W, Q, U, O, R, P, Y, I, C, V, K, D, T}

The Type Switch defines what type of data is to be exported. If the Type Switch is not specified, the program will use the Default variable **I**, which will export **I**nventory Database records. Other variables are determined by the Hotkey (highlighted letter) for each type of data that can be exported using the standard Export function located in the Configuration Manager (e.g., **TYPE=S**, to export **S**ales Invoices).

Sort={1, 2, 3, 4, 5}

The Sort Switch may be used to specify how exported records will be sorted. This option is only available for the following export Types: **I**nventory, **C**ustomers, **V**endors, and **C**lerks. Transaction and payment Types are sorted by date. The default Sort setting is always **1**, which represents the first option in the Sort By list in the standard Export function (e.g., for Inventory, the default is Description). To specify a different Sort By option, enter the line number for the desired option as it appears in the Sort By list (e.g., for Stock Number, enter **SORT=3**).

File=EXPORT.TXT

The File Switch is used to specify the name of the file to which data will be copied. This must be a valid DOS file name and can include the drive and directory path to which the file will be created. The default file name is **EXPORT . TXT**, and is the same as the Output File (or Invoice Totals) setting in the standard Export function.

File2=DETAIL.TXT

The File2 Switch is used to specify the name of the file to which transaction line item details will be copied. This must be a valid DOS file name and can include the drive and directory path to which the file will be created. The default file name is **DETAIL . TXT**. This is the same as the Line Items setting in the standard Export function.

The File2 option is used only when exporting transactions. Header information (e.g., Invoice Number, Date, Customer, Totals, etc.) is copied to the File specification. Line item details (e.g., Stock Number, Description, Quantity, Price, etc.) are copied to the File2 specification. The transaction number (e.g., Invoice Number) is included in both files so that header and line item data can be correlated between the two files.

Warning: *If the specified File or File2 already exists with the same name, it will be deleted and the initial information in it will be lost. **Keystroke** will not prompt you before deleting.*

Start=date {or DATA}

The Start Switch is used to specify the beginning of a range of data. This will be in terms of either a date if you’re exporting transactions, or an alphanumeric starting point (based on the specified Sort By option) if exporting a Database (Inventory, Customers, Vendors, Clerks, Departments, or Item Categories). When exporting a Database, the default Start setting is the first record in the Database based on the Sort By option.

When exporting transactions, the default Start setting is this morning at 0:00 am. A different date can be entered using any of the accepted **Keystroke** date formats. The special Midnight option (**START=M**) may also be used to specify the beginning of either today or yesterday, depending on when the Export function is executed. If run before noon, the Start date will be Midnight yesterday morning, so yesterday's activity will be included. If run after noon, the Start date will be Midnight this morning, and will include today's activity only.

StartTime=time

When exporting transactions, the default Start Time is 0:00 a.m. A different time can be entered using any of the accepted **Keystroke** time formats.

End=date {or DATA}

The End Switch is used to specify the end of a range of data. Just like the Start Switch, this will be in terms of either a date for exporting transactions, or an alphanumeric ending point for exporting a Database. When exporting a Database, the default End setting is the last record in the Database based on the Sort By option. When exporting transactions, the default End setting is this evening at 11:59 p.m. (Midnight tonight). The special Midnight option (**END=M**) may also be used to represent the nearest Midnight to the current time.

EndTime=time

When exporting transactions, the default End Time is 11:59 p.m. A different time can be entered using any of the accepted **Keystroke** time formats.

Filter={2, 3, 4, ... condition }

The Filter Switch may be used to specify a Filter Field and Condition. The available Filter Fields and Conditions vary for each Type of export, and can be viewed in the standard Export function. The Filter Switch is only available for the following export Types: Inventory, Customers, Vendors, Clerks, Departments, and Item Categories. The default Filter setting is **None**.

To specify a Filter, enter the Field line number followed by the Condition (enclosed in quotes). To determine the Field line number, go to the Filter Field list for the desired Export Type in the standard Export function and count which line the field is on (None = line number 1). For example, to export only Inventory items with a Price greater than \$100.00, enter **FILTER=8">100"** (no space between the Field number and condition).

Tab=ON

The Tab Switch determines whether the data fields in the ASCII file will be separated by Tabs or Quotes and Commas. If the **TAB=ON** Switch is specified on the Export command line, data will be copied to a Tab Delimited ASCII file. Otherwise, the default ASCII Quote/Comma Delimited file format will be used. If the program you're importing into supports Tab delimited files, you should use this type since it is faster and more reliable. The **Keystroke** Importer automatically detects and supports Tab delimited files.

EXAMPLES:

To export all Inventory items to EXPORT.TXT, enter the following command at the DOS prompt in the **Keystroke** program directory:

```
KSEXPORT 
```

If it is before noon, to export all of yesterday's Sales Invoices to the file EXPORT.TXT (for the headers) and DETAIL.TXT (for the line items), enter:

```
KSEXPORT TYPE=S START=M END=M
```

To export all Purchase Invoices for the year 1997 to the same file names as above, enter:

```
KSEXPORT TYPE=U START=01/01/97 END=12/31/97
```

To export all Customers who owe you money in the 90802 zip code to the Tab Delimited file MONEY.TXT enter:

```
KSEXPORT TYPE=C SORT=5 FILE=MONEY.TXT START=90802  
END=90802-9999 FILTER=29">0"
```

GL-Link from DOS Prompt GLLINK.EXE

GL-Link is a popular Custom module that can be used to transfer sales and AR data from *Keystroke* to the General Ledger of a third party accounting system. The GL-Link module can be run directly from the DOS prompt (without user intervention). Similar to the ability to run Report Cues directly from the DOS prompt, the GL-Link function can be easily incorporated into automated procedures such as polling scripts, batch files, or other programs.

Running GL-Link from DOS

To run the GL-Link module from the DOS command prompt, you must first be sure that the GL-Link module is properly configured for your use. In other words, don't attempt to run GL-Link directly from DOS until it is otherwise working properly. A complete description of how to set up and use the GL-Link module is included in the Custom Modules section of the *Keystroke* User Guide.

DOS Commands for GL-Link

To run the GL-Link module directly from the DOS command prompt (e.g., polling scripts, batch files, or other programs), use the following syntax.

```
GLLINK START=date STARTTIME=time END=date ENDTIME=time /NOP  
[Other Switches]
```

The program will assume default settings (as described below) for all Switches that are not specified. All other aspects of the GL-Link setup are used as specified in the GL-Link module Setup functions (e.g., Link To, Parameters, Posting Accounts, etc.).

Using Optional Switches

The Start, End, StartTime, and EndTime Switches are used to specify the Start and End Date/Time for the GL-Link transfer. If any of these Switches are not specified, the program will assume the normal defaults for the GL-Link module. Otherwise, these Dates can be specified using the same syntax as described previously for Report Cues.

If the Start Date is not specified, the default Start Date will be equal to the End Date from the previous GL-Link transfer. If the End Date is not specified, the default End Date will be the current system Date/Time. This logic will ensure that no time period is omitted or overlapped.

The /NOP Switch (stands for “No Pause”) is used to tell **Keystroke** not to confirm the overwriting of the file that the data is written to. This Switch should almost always be used if running from an automated procedure.

Other Switches

The GL-Link command may also include any other applicable Startup Switches that are available to the **Keystroke** program. These are described at the beginning of this Appendix; however, the most typical Switch you may need for the GL-Link module is /D to specify an alternate Data subdirectory.

Importer from DOS Prompt *IMP.EXE*

The Importer is a popular Custom add-on module that can be used to transfer Inventory, Customer, and Vendor database information into **Keystroke**. The Importer module can be run directly from the DOS prompt (without user intervention) by executing the command IMP.EXE. Similar to the ability to run Report Cues directly from the DOS prompt, the Importer function can be easily incorporated into automated procedures such as polling scripts, batch files, or other programs.

A list of the data fields and what type of data they may contain is located in the file EXPORT.DOC, located in the DOC\ subdirectory. You can view this file using any text editor.

All functions of the Importer module (except for setup) may also be accomplished using this stand-alone program. To run the Importer directly from the DOS command prompt use the following syntax:

```
IMP /S SetupFile [/I, /C, /V] [/M, /A, /R] [OtherSwitches]
```

Items in brackets ([]) are optional Switches (described below). Do not include the bracket characters in your actual command line. The program will assume default settings for all Switches that are not specified.

/S SetupFile

The /S Switch is used to specify the Importer setup file. The specified file must be located in the FORMS\ subdirectory. The default file KSIMPORT.KSI will be used if this Switch is not used.

/I, /C, /V Database

Which type of database records are being imported is specified by using the /I, /C and /V Switches (Inventory, Customers or Vendors respectively). Inventory is the default option and will be used if the /C or /V Switch is not entered. Use only one of these switches per import.


/M, /A, /R Import Method

How the records are imported is specified by using the /M, /A and /R Switches (Merge, Append or Replace respectively). The default option is Merge, and will be used if the /A or /R Switch is not entered. Use only one of these switches per import.

Other Switches

The Importer command may also include any other applicable Startup Switches that are available to the **Keystroke** program. These are described at the beginning of this appendix; however, the most common Switch you may need for the Importer module is /D to specify an alternate Data directory.

Summarize History **SUMMARY.EXE**

The SUMMARY.EXE command is used to generate the data files used by the  Show Summary function. This program can be launched from within **Keystroke** by selecting the Summarize History function on the Special menu in the Database Manager. It can also be run directly from the DOS prompt. (Actually, even if you launch it from within **Keystroke** the function first exits the **Keystroke**, program and then runs the SUMMARY.EXE program).

If you are planning on using the Show Summary function regularly, we highly recommend that you automate the generation of the summary data files by including this command in a batch or scheduling program so that it will be run consistently and without your having to remember to do it. If you need assistance in setting this up as a semiautomatic batch process (batch program), or a fully automatic process using a scheduling program, please consult your local Authorized **Keystroke** dealer.

The Summary program generates separate data files for each Database, Period, and Date Range you request. Using optional Startup Switches, you can specify which Database, Period, Start and/or End dates for which you want to generate files. The program will use default settings for any information you don't specify. The default settings for each Startup Switch are noted later in this section.

To run the Summary program using the default settings, enter the following command at the DOS prompt in the **Keystroke** program directory:

```
SUMMARY 
```

This command will generate Summary Totals for Inventory, Customers, Vendors, and Clerks for Daily, Weekly, and Monthly periods. The program will work backwards in time (beginning with the previous day), and will generate Totals for the past 7 Days, 5 Weeks, and 12 Months, or until it reaches a Summary Data file previously generated for one of these Periods. Since the Summary program does not regenerate files for Periods that already exist (unless a Start date is specified), if you run the program every night it will only create those files that are necessary (i.e., new Periods).

If you wish to generate Summary Totals for one specific Database (e.g., Inventory), for a specific Period (e.g., Monthly), or for a specific Date Range, simply add the appropriate Startup Switches after the Summary program command. For example:

```
SUMMARY database=I period=M start=01/01/95 
```

This command will generate Monthly Totals for Inventory, from the beginning of 1995 through today's date. Since a Start date was specified, the program will generate Summary Totals for every Month back to January 1995 (instead of just 12 months as would be if no Start date were specified). Since the Start date is used, even if Summary Data files already existed for some of those months, the program will regenerate the files for those months. This may be necessary if you frequently edit old transactions.

Note: *There are a variety of full-featured scheduling programs available from third-party software vendors. System Agent, which is part of Microsoft Plus! for Windows95 is one option. Another simple option is to use **Keystroke's** Program Scheduler described later in this Appendix (KSSCHDLR.EXE, see the KSSUM.BAT file for an example).*

Startup Switches

As previously mentioned, all Startup Switches are optional. Startup Switches can be specified in any order.

/D path

The Directory Switch may be used to specify an alternate Data Directory. This Switch must be used if your data files are not located in the default Data Directory (e.g., **/D STORE1**).

Period={D,W,M}

The Period Switch may be used to specify an individual Period (**D**aily, **W**eekly, or **M**onthly). If this Switch is not used, the program will generate files for all three periods by default.

Database={I,C,V,K}

The Database Switch may be used to specify an individual database for which files should be generated (**I**nventory, **C**ustomer, **V**endor, or **C**lerk). If this Switch is not used, the program will generate files for all databases.

Start=date

The Start Switch may be used to specify the oldest date for which Summary Totals should be generated. If this Switch is not used, the program will generate files for the previous 12 Months, 5 Weeks and 7 Days, or until a previously generated Summary file is found. Dates can be specified in any format acceptable in any Date field in the **Keystroke** program (e.g., **start=06/01/97** or, to start one week ago **start=-1W**).

End=date

The End Switch may be used to specify the latest date for which Summary Totals should be generated. If this Switch is not used, the program will generate files up through the previous midnight. **End=M** will go through the closest midnight, so if the Summary program is run in the evening it would include the current day.

/Pause

The Pause Switch will cause the Summary program to pause before creating each file. Settings used to generate the file will be displayed. Since this will require user intervention, this Switch should not be used when running the Summary program unattended through an automatic scheduling program.

/MoveOld

The MoveOld Switch may be used to remove older Summary data files from the Data directory so they will not be accessed when you press **CTRL F5**. If no other Switches are used, the MoveOld Switch will move all Summary data files older than the default Periods (7 Daily, 5 Weekly, and 12 Monthly) from the Summary subdirectory to the OLD\ subdirectory beneath the SUMMARY\ subdirectory.

Summary Data Files

Data files created by the Summary program are named according to the following syntax:

dpyymmdd.DAT Example: **ID970101.DAT**

Where “**d**” represents the Database (**I**nventory, **C**ustomer, **V**endor, or **C**lerk). “**p**” represents the Period (**D**aily, **W**eekly, or **M**onthly). ‘**yymmdd**’ represents the **Y**ear, **M**onth, and **D**ay of the end of the respective Period.

Using the MoveOld Switch

The MoveOld Switch is a special Switch that can be used to automate the management of your Summary data files. When this Switch is used, the Summary program will not generate any new data files. Instead, it will simply move older data files from the Summary subdirectory to the OLD\ subdirectory (e.g., **C:\KEYSTROK\DATA\SUMMARY\OLD\ID970501.DAT**).

By default (if no other Switches are used) the MoveOld Switch will move all Summary data files older than the default Periods (7 Daily, 5 Weekly, and 12 Monthly) from the SUMMARY\ subdirectory to the OLD\ subdirectory. The primary purpose of doing this is to eliminate older Summary Totals from being displayed when you press **CTRL F5**. If you do not periodically manage these files (especially Daily files), the system will eventually have too many files to be practical, which will slow down the **CTRL F5** function, and could also result in system errors.

To run the Summary program with the MoveOld Switch, enter the following command at the DOS prompt in the **Keystroke** program directory.

```
SUMMARY /moveold 
```

If you are using a Batch program or other automatic scheduling software to generate Summary Totals, this command can be added to the same routine so it will run on the same periodic basis. If you have no intention of ever using the old Summary files (e.g., for reports), you may also want to follow this command with the DOS Delete command (e.g., **DEL C:\KEYSTROK\DATA\SUMMARY\OLD*.DAT**) in order to permanently remove the old files and free up hard disk space.

The Start and Period Switches can also be used to define which files will be moved. When used in conjunction with the MoveOld Switch, the Start Switch may be used to specify the date from which the program will count backwards from in order to determine which files to move (e.g., **START=-1M** would only move files older than the default Periods as of one month ago). In other words, files before the previous month plus the default Periods would be moved. The Period Switch can be used to move a specific type of Period files (e.g., **PERIOD=D** would move only Daily files).

If you would like to maintain more than the default Periods, it may be necessary to run the Summary program with the MoveOld Switch multiple times. A typical Batch program might include the following commands.

```
SUMMARY database=I  
SUMMARY /moveold period=D  
SUMMARY /moveold period=W start=-1M  
SUMMARY /moveold period=M start=-1Y
```

These commands would generate Summary Totals for Inventory for the default periods, and would move older Summary data files leaving only the most recent 7 Daily files, 9 – 10 Weekly files, and 2 years worth of Monthly files.

Auto-Production from DOS Prompt PROD.EXE

Production is a Custom add-on module that is used to help manage Inventory Quantities On Hand for items (finished goods) that are either produced from individually purchased components, or broken down from products purchased in bulk units (e.g., cases). The Auto-Produce function included in the Production module can be run directly from the DOS prompt (without user intervention). Similar to the ability to run GL-Link directly from the DOS prompt, the Production function can be easily incorporated into automated procedures such as batch files or other programs.

To run the Auto-Produce function directly from the DOS command prompt (or batch program), use the **PROD** command. This will execute the stand-alone utility program **PROD.EXE**, which performs the exact same procedure as the Auto Produce function located on the Template menu in the Production module. All standard Startup Switches available to the **Keystroke** program are also available for use with the Auto-Produce command. For complete instructions on how to set up and use the Production module, please refer to the Production Module documentation in the Custom Modules appendix.

StoreLink Inventory Update STRUPDTE.EXE

The optional StoreLink Custom add-on module is part of the MultiStore package. This module contains an Update Inventory function (located on the Store menu). This function updates to Inventory records from the master Inventory files to all the other stores data files. This function can also be run directly from the DOS prompt using the STRUPDTE.EXE program.

There are no special Switches for this program, although all the standard Start Up Switches (like /D) can be used. Similar to the GL-Link program, the parameters have to be set up beforehand from within the StoreLink module.

Stand-Alone Utilities

The following utilities cannot be run from within the **Keystroke** program; they can only be accessed via the DOS command prompt.

Show Document SHOWDOC.EXE

The Show Document program (**SHOWDOC.EXE**) is a simple utility which can be used to view any text file. It is included so that you can easily view the Help screens which are displayed whenever a stand-alone program is run with the /H or /? Switch. These files are located in the DOC\ subdirectory below the program directory.

To use this utility, enter SHOWDOC followed by the name of file to be viewed (e.g. **SHOWDOC DOC\KSEXPORT.DOC**). To leave the program, press .

4.0 Conversion **CONV40.EXE**

The 4.0 Conversion program (**CONV40.EXE**) is a simple utility which can be used to manually convert an individual data file from version 3.2x to version 4.0x.

To use this utility, type **CONV40** followed by the name of file to be converted (including its path and extension) and press **ENTER**. The following files can be converted using this program. The first five letters of the file name you want to convert must match the first five letters of one of the files below:

PRCFRM Price Formulas

TAXFRM Tax Formulas

TRNPMNTS Sales Payments

PORPMNTS Purchasing Payments

FORMS Print Buttons

TIMECLK Time Clock Entries

REPORTS Report Cues

KSTRN* Transactions (* indicates the transaction file number (e.g., 001, 002, etc.))

SCRNS Database Screens

KSINV Inventory

Setup **SETUP.EXE**

The Setup utility is an executable program which reads in configuration parameters from a file with the extension .SET. It then updates the Configuration Data files with the new settings.

The .SET files are text files which are stored in the FORMS\ directory, and can be edited with any text editor. The easiest way to use the Setup utility is to start with an existing file and make changes as needed. To run the Setup utility, type **SETUP**. The program will display a list box of the available files. To update **Keystroke** with the settings stored in a file, simply highlight the file and press **ENTER**. Pressing **INS** will allow you to save the file with a new file name with the current program configuration settings. Press **F3** to update an existing file; however, this method will only update devices already in the selected file (i.e., if the file contains only Cash Drawer parameters, only Cash Drawer parameters will be written to it).

The .SET file does not need to contain every single supported parameter; those not included will simply not be changed. This allows you to have one file for Cash Drawer parameters, one file for Pole Display parameters, etc. You can then select the individual files to copy those parameters into Keystroke.



This program can be run using any of the standard Keystroke startup switches. It can also be run without interaction by using the following syntax:

SETUP FILE=filename where *filename* is the name of the file containing the settings you want to copy into **Keystroke**.

Program Scheduler *KSSCHDLR.EXE*

The Program Scheduler (**KSSCHDLR.EXE**) is a simple utility which can be used in conjunction with DOS Batch programs to cause other programs to automatically run at a preset time of day. Typical uses include MultiStore polling procedures and the new Summarize History feature described earlier in this Appendix. It can also be used to start any program (not just **Keystroke** related ones), including backup and disk management functions.


When this program is run, it will continuously display a screen blank message until the specified time is reached. (Essentially causing the batch program to wait for the specified time before it can continue). It will then exit back to the DOS prompt (or the program that launched it). This will then allow the rest of the programs in the batch file to continue.

While the Program Scheduler is running, you can press  to bypass the Scheduler and continue the Batch program immediately, or any other key to cancel and abandon the Batch program. If any key besides  is pressed, the program will set the DOS ErrorLevel to 1. Your batch file should test for this error level and abort if it has been set.

A Batch program designed to use the Program Scheduler should conform to the format:

```
KSSCHDLR.EXE %1  
IF ERRORLEVEL 1 GOTO END  
{program commands}  
:END
```

Where *{program commands}* represents the body of the Batch program and can be as many lines as necessary. For example, you could substitute the four Summary commands shown in the previous section. A sample batch file (**KSSUM.BAT**) is located in your **Keystroke** directory. The DOS command to run this batch file and have it wait until 1:00 a.m. would be as follows:

```
KSSUM 1:00a 
```

The command line must include the desired preset time of day, which can be specified in any format acceptable by **Keystroke** Time entry fields (e.g., **1** or **1a** would both represent 1:00 a.m.). If a time is not specified, KSSCHDLR.EXE will set the DOS ErrorLevel to 1.

The message displayed on the screen while the scheduler is running can be changed to say anything you want. It is controlled by the file KSSCHDLR.TXT (located in the **Keystroke** directory). This file uses the same format as the SCRNSAVE.TXT file. Please see the Screen Saver section later in this appendix for details on the format for this file.

Recurring Invoices/Recovery Merge *RECMRG.EXE*

Keystroke includes a special utility program titled RecMerge that can be used to copy sales transactions from one set of data files to another. RecMerge is a single stand-alone program (RECMRG.EXE); however, it may be used in a number of different ways, depending on your needs.

RecMerge may be used to automate the task of entering Recurring Invoices (e.g., for monthly service contracts, membership dues, or long term rentals). RecMerge may also be used to merge transaction data from one machine to another. This could be a way of working around a temporary network failure, or it might be a way of consolidating transactions entered away from the primary system (e.g., a portable computer).

Whatever purpose you might have for using RecMerge, it essentially works the same. It copies sales transactions from one **Keystroke** data file to another, and at the same time updates Customer and Inventory records to correspond with those new transactions. More specifically, RecMerge reads the transaction data file **KSTRN001.REC** and merges its contents into the standard transaction data files (normally **KSTRN###.DAT**). RecMerge can be told to assign a specific Date and a new Invoice Number to each transaction, as if it were being entered manually. Unlike the similar MultiStore utility StoreMerge, RecMerge handles only sales transactions (Invoices, Sales Orders, Lay-aways, Quotes, and their respective Payments); it does *not* merge purchasing transactions, Variances, or new or changed Customer records.




Note: The RecMerge utility is best used only with systems that are otherwise fully set up and operational. Because of the way RecMerge works, you should get all other aspects of your system and databases set up before attempting to set up RecMerge.

Setting Up RecMerge for Recurring Invoices

To use the RecMerge utility for Recurring Invoices, you'll first have to set up a new set of **Keystroke** data files to enter the Invoices into. There are actually several ways this can be done, but for simplicity's sake we will discuss only one such procedure. In this method we will create a second copy of your normal data files and give them the extension **.REC**. We will then initialize new transaction data files.


To begin, you will need to exit the **Keystroke** program and get to the DOS command prompt in the Program Directory (e.g., **C:\KEYSTROK>**, or **F:\KEYSTROK>** on networks). If you are not completely comfortable with DOS commands, you should contact your local **Keystroke** dealer for assistance.

You are now ready to create a new set of **Keystroke** data files that will be used to set up your recurring transactions. Copy your default data files to the **.REC** extension, then delete the transaction data files with this extension, and start the **Keystroke** program with the **/E** Switch, which tells it to address the alternate data files. To summarize these commands:

1. Type **COPY DATA*.DAT DATA*.REC** 
2. Type **DEL DATA\KSTRN*.REC** 
3. Type **KEYSTROK /E REC** 

This sequence will launch the **Keystroke** program where you can enter all the sales transactions that will be recurring for the same period (e.g., monthly). The first time you go into the program, the file **KSTRN001.REC** will be initialized.

Everything in **Keystroke** will work identically to what you're used to, except there will be no transaction history. Enter a single Invoice with the appropriate method of payment (a Credit Payment Type such as "On Account" is usually appropriate) for each one that will be recurring. Do not be concerned with Invoice Numbers or Dates, since they will be reassigned as the transactions are merged into your standard data files. After entering transactions that are to be recurring, exit the program.

You may add, edit or delete these transactions any time by typing **KEYSTROK /E REC**  from the DOS prompt in the KEYSTROK\ directory, or by using the Extension function in the Configuration Manager.

Setting Up RecMerge for Network Recovery

RecMerge can be used to merge in sales transaction entered on another computer (such as after a network failure, or when a laptop computer is used on a temporary basis). In this case, you will need to copy the transactions file(s) to be merged onto your main computer to the file KSTRN001.REC (be careful not to overwrite any of your other transaction data files). To merge more than one transaction file, run RecMerge separately for each file. For more detailed instructions, see the section titled Fail-Safe Network earlier in this Appendix.

How to Merge Transactions Using RecMerge

RecMerge only merges transactions from the file KSTRN001.REC. If you are merging transactions entered separately from your standard data files due to network failure or other reason, you'll first have to copy or rename the source transaction file (e.g., KSTRN001.NEW) to the filename KSTRN001.REC. If you are using RecMerge to batch in Recurring Invoices and you followed the instructions above, this file will be ready to go.

To run RecMerge, type:

```
RECMRG [DATE={date},0] [/ADD][/NOP][/R][/M{Path}] 
```

The **DATE={date}** parameter may be used to specify the date that is assigned to sales transactions as they are merged in. The date can be entered in any acceptable **Keystroke** format (e.g., MM/DD/YY). If this Switch is omitted, all transactions will be assigned the current system date (today's date). If **DATE=0**, merged transactions will keep their original dates.

The **/ADD** Switch may be used to merge in transactions "as is", so that Invoice Numbers and dates will not be updated. This would generally be used for merging transactions from another computer (such as after a network failure) and *not* for Recurring transactions. The **DATE** and **/ADD** parameters should *not* be used together; use one or the other.

The **/NOP** Switch may be used to run the RecMerge program with "no pauses" for unattended use. If this Switch is used, the program will not stop to tell you what it is about to do or what it has already done.

If you are using the Network version of **Keystroke**, the **/R** Switch may be used to assign a specific Register Number to the transactions that are merged in. This can be especially useful if you want to use a different Number than on your actual Registers, so that you can later run reports Filtering on this Register Number to see only merged transactions.

The **/M {path}** Switch may be added if the KSTRN001.REC file is located in a subdirectory other than the Data Subdirectory. Where **{path}** represents the DOS path (e.g., **C:\KEYSTROK\MONTHLY**).

All other standard Startup Switches may also be used with the RecMerge program.

Print Labels from Invoice Form FORM2LBL.EXE

One of **Keystroke**'s most powerful features is the ability to print customized forms such as Invoices, Statements, and Purchase Orders. As with any printed output in **Keystroke**, forms can also be printed to a file instead of a printer. The file created is called a text or printer file (depending on whether or not print codes are included). This text file can then be sent to a printer, pulled into a word processing program (or any other program that supports text files), or it can be treated as special ASCII fixed length data file. As such, it is a fairly simple matter to read in this text file and use the data to: Add new information to the form not available within **Keystroke**; Add in special fields and calculations; Record the data for use by another program to generate special reports or graphs; Send this data on to another computer system (e.g., a purchase order can then be sent electronically to your vendor); Print out tickets when certain items are purchased; or Reformat the file to produce a new type of form not supported by **Keystroke**, (e.g., label stock for pick lists printed as labels).

The **Keystroke** FORM2LBL.EXE program is designed to be used for the last case listed above. We have included this program and its source code – FORM2LBL.BAS (written in Microsoft's QuickBasic) as an example of what can be done with a form printed to file and how to do it. The program will search for any file with the extension .FRM. If a file is found, the program will process it.

***Note:** You can make the Invoice Number be part of the file name by putting an "*" in the file name field of the printer setup box. This allows you to print forms from **stroke** even if your*

*Mode command so **Key** will keep adding each form to the end of the file instead of replacing it).*

be run as a separate task), and an executable mode (which is called by the DOS Command on the **stroke** printer setup screen). Again, this is just a sample. The same type

macro language included with most word processing and spreadsheet applications) and run under any operating system (since it can read the text file created when the form is

This program was originally written for a store that wanted to print a label for each item sold and to include the Customer and Invoice information on each label. This was used to **Keystroke**

Invoice form, this program was used to copy the Customer's name from the header to each line item in the body.

header section of a form to be placed within each line item (i.e., the body section). It does this by reading in a text file which contains the value for special fields "Token

in the body section "Tokens" with the value of the Token. The text file name must have the extension .FRM. The program will search for any file with the extension .FRM, read

a single character between two ampersands, followed by the value of the Token:

The above statement defines the Token **&C&** to be the name “Joe Smith”. The header section must end with a line that starts with **&&**. In the body section, include the Token wherever you want its value to appear. For example:

Item purchased by: &C&

The above will cause the program to replace **&C&** with **Joe Smith**.

The following Switches are supported:

/R Resident mode: *The program will continuously loop searching and processing files with the .FRM extension*

/S Stop: *The program will pause before exiting*

/P Printer output: *The converted file will be sent to the device/filename following the /P*

Note: *If you are using **Keystroke** on a network, you may want the .FRM files and the FORM2LBL program to be stored on your local machine. Otherwise, it's possible that a different machine will see your form and print it to its printer instead of to your printer.*

There are several steps to using this program. First of all, you need a form file that is in the format you want (e.g., one that will print each line item on a separate label). This form should include the Token information as described above.

Note: *The creation/editing of form files is not available within the **Keystroke** program. Please contact your Authorized **Keystroke** dealer for assistance in creating/editing custom form files.*

Once you have specified the form file, you will need to set up a printer for **Keystroke** to print to (it will actually be a file name), and then set up a Print Button to be used whenever you want to print the form. The printer setup should be done by adding a new printer (e.g., Form Labels) and setting that printer to use the same printer driver the real printer uses, but set the port to File. In the Output File box enter the name of the form (e.g., *.FRM so it will be the InvoiceNumber.FRM).

The easiest way to have FORM2LBL process the form file after it is printed is to use the DOS Command field within the Printer Setup. By entering FORM2LBL (plus the /S and or /P Switch if needed; do not use the /R Switch) in this field, the program will call (or run) the FORM2LBL program every time a new form is printed.

Another way to run FORM2LBL is to have it running continuously. This is done by running it with the /R Resident Mode Switch. In this mode, the program will constantly watch for the appearance of a file with the .FRM extension. As soon as one appears, it will process it. The advantage to this method is that **Keystroke** does not have to unload itself, run FORM2LBL, and then reload itself each time a form prints. However, since both **Keystroke** and FORM2LBL are DOS programs, you cannot run them at the same time. If you have a multitasking operating system (e.g., Microsoft Windows (3x, 95, or NT)), you can run each program in a separate DOS window. In this case, the operating system's multi-tasking capabilities will allow both programs to run at the same time.

If you are using a network, you can also run FORM2LBL on a different computer. The only requirement is that both your computer and the one running FORM2LBL must be able to share the same drive (i.e., you want to be able to print the form to that drive and the FORM2LBL program must be able to read from it).

Format Qty Column on Invoice Form FORMQTY.EXE

This is another example of a simple utility used to modify the way forms are printed. It is used in the same way as the FORM2LBL program described above, except that it can cause any numeric field to print with a fixed number of decimal points (e.g., quantities always show 3 decimal places (1.000 instead of 1)).

Any field that you want to print with a fixed number of decimal points should be preceded by the Token characters **&Q#** (where **#** indicates the number of decimal points). For example:

&Q2	1&	will be changed to	1.00
&Q2	1.2&	will be changed to	1.20
&Q2	99.999&	will be changed to	100.00

Note that the token characters are thrown away, so when creating a form, it is not necessary to count the spaces taken up by the **&Q#** or the last **&**.

Import Accounts Receivable Balances IMPORTAR.EXE

This program imports Accounts Receivable Customer Balances into **Keystroke** by creating a Balance Forward Invoice for the balances contained in the ASCII delimited file CUSTBAL.TXT. The first field in this file is the Customer Code (unless changed by a Switch); the second field will be the total balance. Any other fields will be balances for previous periods. The balances from previous periods are optional; the amount posted to the current period will be the total minus these other fields.

Keystroke uses an Open Item method for handling Accounts Receivable. This means that a Customer's balance must be supported by transactions (Invoices or ROA payments) with an amount due on them. In order to import a balance for a Customer, the program must create an Invoice to support that amount due. This is done by automatically entering a new Invoice for the Customer with a nontaxable non-Inventory line item (Stock Number = 0) called Balance Forward, with the price set to an amount due. The date on this Invoice will be ending date for the appropriate period and will have an On Account payment type attached to it.

To run the Import Account Receivable Balances utility from the DOS command prompt, use the following syntax.

```
IMPORTAR [/F {FileName}] [/C {C}] [P0=date] [P1=date]  
[P2=date] [P3=date]
```

Text in braces (**{ }**) represents required variables. Items in brackets (**[]**) are optional Switches (described below). Default settings are used for non-specified Switches when necessary. Do not include the bracket or brace characters in your actual command line. The entire command line must be entered on one line (syntax above is displayed on two lines due to space limitation only).

Note: *Since this program alters your **Keystroke** data files, always be sure to make a backup!*

/F FileName

The /F FileName Switch is used to define the name of the file to be read in. The default is CUSTBAL.TXT

/C Customer Field {C, N, M, P, Z}

The /C Customer Field defines which fields the Customer should be looked up by. After the /C, enter the letter code that defines which field to use (Customer Code, Customer Number, Name, Phone, and Zip). If this Switch is not specified, the program will use the Customer Code field.

P#= Period Dates

The P#= Switches are used to define the dates that will be used for each previous period. If these Switches are not used, the period dates will default to the end of each previous month. The period dates are used to assign the Invoice dates (e.g., if today is 3/15/97, the program will default to assign the Invoice with a current balance to be 2/28/97. The Period 1 balance would be 1/31/97, etc.).

Note: While this program was designed to be used when first switching over to the Keystroke program, it can be used to create Balance Forward Invoices for some of your Customers on an ongoing basis (an alternative to using the RECMRG function).

Import Alternate Inventory Codes IMPALTIC.EXE

The program IMPALTIC.EXE is used to import Alternate Inventory Codes from an ASCII delimited file (either quote/comma or tab delimited). The default file name is ALT_CODES.TXT. The first field in each line of this file is the Alternate Code and the second field identifies which Inventory item to attach it to (i.e. Stock Number, Product Code, etc.).

The program will read each Alternate Code and then search for the Inventory item to attach it to. If the Inventory item is not found, the Alternate Code is not added. The program will also check to see whether the code already exists for the same Inventory Item; if it does, it will not be added. After completing the Import, the program will display the number of codes read from the file, the number added, the number skipped (since they already existed), the number of bad Codes (Inventory item could not be found), and the number of duplicate Codes (same Code for more than one Inventory item) that were added.

To run the Import Alternate Codes utility from the DOS command prompt, use the following syntax:

```
IMPALTIC [/F {FileName}] [/C {N}]
```

Text in braces ({}) represents required variables. Items in brackets ([]) are optional Switches (described below). Default settings are used for non-specified Switches when necessary. Do not include the bracket or brace characters in your actual command line.



/F FileName

The /F FileName Switch is used to define the name of the file to be read in. The default is ALT_CODES.TXT.


/C Code Field {N, D, C, L, M}

The /C Code Field defines which field the Inventory item should be looked up by. After the /C, enter the letter code that defines which field to use (Stock Number, Description, Product Code, Class, Manufacturer). If this Switch is not specified, the program will use the Stock Number field.

Bike-a-Log On Disk Link BODLINK.EXE

The BODLINK.EXE utility is included with the optional Importer module. Bike-A-Log on disk is an electronic catalog of parts for the bicycle industry. This program contains a list of over 160,000 items. Within the Bike-a-Log program, you can Mark (by pressing  ) which items you want copied into **Keystroke**. This utility is used to extract those Marked items into an ASCII delimited file which can then be imported by the **Keystroke** Importer module. Please see the on-line help (**BODLINK /H** or **EDIT BODLINK.DOC**) for more information.

Delete Utility DELUTIL.EXE

The Delete Utility is a standard **Keystroke** module that can only be accessed from the DOS prompt. This program is essentially a data file repair tool that should only be used with the help of **SBS**. It is used to manually make changes to your data files that cannot be done from other parts of the **Keystroke** program. To run this program, type **DELUTIL**  at the DOS prompt, followed by any Startup Switches you normally use to start **Keystroke**.

Delete Menu

The Delete menu is used to mark records for deletion, but it does not actually delete the records. You must then go back into the **Keystroke** program and Pack the appropriate file in order to complete the deletion process (records marked for deletion using this utility will continue to show up on reports until the Pack function is run). From the Delete Menu, you may choose from the following options:

Payments Received Function *mark payments received in the Sales Manager as deleted*
Payments Paid Function *mark payments paid in the Purchase Manager as deleted*
Back Orders Function *mark Back Orders from the Sales Manager as deleted*
Open Orders Function *mark Open Orders from the Purchase Manager as deleted*
Credit Memos Function *mark Credit Memos as deleted*
Audit Records Function *mark Audit records as deleted*

Update Menu

The Update menu is used to recreate subsidiary data files, recalculate Inventory quantities On Layaway and Allocated, make global changes and/or repairs for the Accounts Receivable module, and to adjust transaction file splits.

Credit Memos Function/Open Orders Function

The first two options on the Update menu are used to rebuild the Credit Memo and Back/Open order lists. Both of these files are simply lists that can be used for quick reference to find the actual data to be viewed and/or edited (e.g., an ROA payment not fully applied or a transaction not fully shipped/received). Since these are subsidiary lists, the program has the ability to recreate them through this utility. These functions will first delete the file to be recreated and then search your transaction history to recreate the file in the correct position.

Qty Allocated/On Layaway Function

Each Inventory item tracks the quantity that is currently on Layaway or on Back Order (sales transactions where the quantity ordered and the quantity shipped are different). If these numbers ever get out of sync, they can be recalculated by using Update Qty Allocated/On Layaway function. This function will first set the Allocated and Layaway fields on every Inventory item to 0 (zero). It will then search each transaction and update these quantities for all Layaways and Back Orders which are found.

Update Customers Function

The Update Customers function is used to set every Customer's Oldest Open Invoice date back to an earlier date. **Keystroke** uses this date to determine how far back the program should search for Open Invoices for each Customer (when printing statements, reports or entering an ROA payment). This date can be changed for each Customer, one at a time, by using the Finance Edit function in the Accounts Receivable module, but the Update function is used when you need to make sure that the program searches back to a particular date for *all* your Customers. The date should be changed to the oldest date on which any Customer could possibly have an open Invoice. The first time the program reviews a Customer's A/R balance after running this function, it will take longer than usual since it will search further back than it really needs to. After the first time the program reviews the Customer's account, it will reset the Oldest Open Invoice date back to the correct date for each Customer.

ROA/Invoices Function

The ROA/Invoices function is a repair utility that should only be used when instructed to do so by **SBS**. This function will recalculate (and correct if necessary) the amount due on each Invoice. It will then check each ROA payment and make sure that all Invoices have received the correct credit for the payment (it will correct any Invoices that are missing their applied payments). This function records all changes made in the file FIXAR.TXT.

Transaction File Splits Function

The Transaction File Splits function is a repair utility that should only be used when instructed to do so by **SBS**. This function will automatically update file lists, fix transaction file information (if necessary), and change the size and date range of transaction files. To use this utility, specify the starting date and new size for the files you wish to adjust.

The Delete Utility can be run using the /G Switch, which will automatically run the specified function. Functions are specified by entering the highlighted Hotkeys from the Delete Utility Menus (e.g., for Update – Open Orders, type /GPO). Due to security concerns, not all functions are supported by the /G Switch.

Note: *Since this program alters your **Keystroke** data files, always be sure to make a backup!*

ChangeNums CHGNUMS.EXE

A utility program called ChangeNums is available for users who need to renumber Customer and/or Inventory database records. This utility is intended for situations where the database “record numbers” (Customer and Stock Numbers) are approaching their maximum (65,535). This does not overcome the limitation on the number of database records allowed by the system, which is 65,519.

This utility may be helpful to you if your highest Customer Number or Stock Number is approaching 65,535, and there are significant gaps in those record numbers (meaning that there are actually far less than 65,000 records, usually because a large number of records have been deleted). Unless you have a need to add a large number of Customer or Inventory records, you probably don’t stand to gain much from this utility.

You should not use this utility just because you don’t like the way things were numbered or to “fill in the gaps” for records that have been deleted. These numbers are used for tracking purposes only; they do not and should not have any special meaning attached to them.

Keystroke tracks all of its data on a relational basis according to the “record number” for each record in your databases. Therefore, if you were to manually change a Customer or Inventory number, all of its history would be essentially lost (Or worse, it could pick up the history from a different record that used to have that same record number). For this reason, **Keystroke**’s security has been set to not allow you to change Inventory Stock Numbers, Customer Numbers, Vendor Numbers, or Clerk Numbers.

The ChangeNums function is provided as a way to change these numbers when you absolutely have to. This program will renumber your database records starting at the number specified and then counting up one number at a time. The numbers will be assigned in the order in which the records were entered into the database file. After the numbers are reassigned, the program will go back through all of your transaction files (Sales/AR/Purchase/Variance history) and adjust all the information to reflect any changes made by the ChangeNums program.






MultiStore

One legitimate use of this function is if you have several stores that have been operating independently and you now want to consolidate them using the MultiStore features. In order to use these functions, all common database records must use the same record numbers. This program, along with the Export function and Importer Module can be used to get all these numbers in sync between the different stores. The key to this procedure is that ChangeNums will reassign the record numbers based on the order they were added to the database file. You can export the Inventory or Customer database in the order you want them to be renumbered (e.g., sort by Code), then do an Import – Replace. This will cause your database records to be physically entered in the order you want them to be. Then run ChangeNums to make the actual Numbers agree with this new physical order.

If you are positive you have a need to renumber your Customer and/or Inventory records, please call your local Authorized **Keystroke** Dealer or **SBS** for additional instructions and assistance.

Print Unused Stock Numbers NUMBERS.EXE

This program will print out a list of all the available (unused) Inventory Stock Numbers. Like the ChangeNums function, this really should not be used except in extreme cases. As mentioned above, the Stock Number really has no meaning (the program does not care if numbers are skipped), and reusing an old number leaves the possibility of confusing the transaction history of the previous item with that of the new item.

This program is accessed by entering **NUMBERS**  at the DOS prompt. There are no special Switches for this program; however, it does support all the standard Startup Switches. The program will then ask how many numbers you want listed. Enter the desired number (e.g., **100**, and press ). The program will then ask where you want the list printed. Enter  for **S**creen (the list will appear on you monitor),  for **P**rinter (The list will be sent to your LPT1: port), or  for **F**ile (the program will then ask for the name of the file to print to).

Note: *The following three utilities should only be used when instructed to do so by SBS. More complete documentation is available by viewing the help screen for each program (use the /? Switch or view the corresponding .DOC file).*

Check Inventory Script Flag CHKSCRIPT.EXE

Keystroke has an optional Scripting Module available which can be used to define a Script to be run each time a particular Inventory item is sold. Each Inventory item contains a hidden flag which marks whether or not it has a corresponding Script (this speeds up data entry since the program does not have to check for a Script that does not exist). If you manually delete or copy your script files, it is possible that these flags will not be set correctly. The CheckScript program will check each Inventory item and make sure that this flag is set correctly.

Stamp STAMP.EXE

The Stamp utility will stamp the specified files with the current Date and Time by writing a serialized representation of the Date/Time the program was run to the first record of the file. You must specify one (and only one) file name, but that file name can include wildcards to represent several names (e.g., KSTRN*.DAT). The specified size of each file must be at least 2048 bytes and be divisible by 128 (all Keystroke Database and Transaction files meet these parameters). This stamp can then be used by the CheckStamp utility to make sure all files are from the same data set.

Check Stamp CHKSTAMP.EXE

This utility will check to make sure all the specified files belong to the same data set by reading the Date/Time stamp written to the file using the Stamp utility. You must specify one (and only one) file name, but that file name can include wildcards to represent several names (e.g., KSTRN*.DAT). The program will write the results to the file CHKSTAMP.LOG, and will return an ErrorLevel 1 if any files don't match, or an ErrorLevel 2 if any files are not valid.

Check Header Information on Transaction Files *HEADER.EXE*

Each **Keystroke** transaction file contains a header record which stores that file's date range and its corresponding place in the chain of transaction files (see the Transaction Data Files section earlier in this chapter for more information). The Header program can be used to view and edit these header records. To print a list of header information to the Report printer, press **SHIFT** **F10**.

Warning: *Incorrect use of this function can cause permanent damage and loss of data! Consult SBS before attempting any use of this program.*

Remove Duplicate Sales Transactions *REMDUPES.EXE*

There are several functions that can be used to merge transactions into your data files: RecMerge, StoreMerge, and Configuration Manager – File menu – Merge. By using these functions it is possible to accidentally merge the same set of transactions more than once, resulting in duplicate transactions in your data files. The RemDupes program will check for these duplicates and will remove any extra copies of the same transaction. This program uses the same Delete function found in the **Keystroke** program and will thus undo any changes made to your database files by the transaction (i.e., it will put items back into Inventory that were used on the transaction to be deleted). This program will only delete transactions that are identical (same Number, Customer, amount, date and time); it will not delete transactions that were entered manually twice (two manually entered transactions will hardly ever be identical since they will not be saved with the exact same date/time).

Copy Next Number *COPYNUM.EXE*

This program can be used in conjunction with the MultiStore utilities to reset the remote stores' Next Number value. CopyNum will copy the second record (FileData, which holds the Next Number values) from the transaction file KSTRN001.MRG to the Customer data file KSCST.DAT. This program supports the standard Keystroke command line switches, as well as the following special switch:

/M *Merge Directory (the location of the KSTRN001.MRG file)*

Miscellaneous Files

Large Numbers *LRGPRINT.TXT*

This file controls the how the numbers are created (i.e., the “screen Font”) in the Sales Manager when the Detail Box parameter is set to Total (Large Print). When the program starts, it reads the characters from the file LRGPRINT.TXT. If this file does not exist, it can be created by copying LRGPRINT.SCR from the FILES\ subdirectory. **Keystroke** ships with three of these files: LRGPRINT.SCR (the default; uses a block character to display the Font); LRGPRNT2.SCR (uses double line characters, and gives a more defined and a slightly smaller Font); and LRGPRNT3.SCR (the same as LRGPRINT2.SCR, but it has a space between each Font character). To use one of these other files or one of your own, simply copy it to LRGPRINT.TXT in the program directory.

LRGPRINT.TXT is a text file that can be edited using any standard text editor. This file defines what characters are used to make of the Font used to display large numbers. Each individual Font is made up of three lines of characters. Each line can contain as many characters as needed (but anything more than 4 is impractical because the Font probably won't fit on the screen). Each row for an individual Font should use the same number of characters (the first row is used to determine the width of the Font). Each line in the file must begin and end with a quotation mark.

The file should be 42 lines long (i.e., 14 font characters times 3 rows each). The first three lines define the Font used if the program tries to display a character that does not have a defined Font (this should never happen). The next ten Fonts (30 lines) define the number 0 through 9. The last three Fonts are the period, comma, and dash (or minus sign) respectively. The best way to create your own font is to edit one of the existing ones and save it under a different name.

Screen Saver *SCRNSAVE.TXT*

This file controls the appearance of the screen saver feature which runs whenever the screen blank time period is exceeded. The first line in this file controls in what position, using what color and how the rest of the information is displayed. It is entered in the following format: *TimeInterval*, *xPos*, *yPos*, *ForegroundColor*, *BackgroundColor*. Setting any of the parameters (except the *TimeInterval*) to 0 will cause the program to randomly select that value causing the displayed text to move around on the screen and/or to change colors.

The *TimeInterval* is the number of seconds between each change. If set to 0 (zero), no changes will occur. The *xPos* and *yPos* settings control the horizontal and vertical positions (respectively) of the message on the screen. The valid settings are:

TimeInterval=0 to 9999

xPos=1 to 60

yPos=1 to 20

ForegroundColor=0 to 15

BackgroundColor=0 to 8

The standard **Keystroke** defaults for this file are **5,0,0,0,8**. The remainder of the file is comprised of the actual text that will be displayed when the Screen Saver function is running. This text may be any size, up to the limits of the screen display (80 columns by 25 lines). All text will be displayed as is, with the exception of two special variables: **\$D** and **\$T**. These special variables will be replaced by the current system date and time, respectively.

Common Errors and Problem Solving Tips

This section lists the most common problems and error messages that can be encountered using the *stroke* system. Following the description of each error or problem are

are usually hardware related; however, this does not discount the possibility of software related problems. Anyone with any experience working around computers will confirm

It is our goal to minimize the frequency of these problems, as well as the resulting consequences when such difficulties do occur. It is therefore very important that we

The best and fastest way to resolve a problem is to first write down any error messages or other details which may help in resolving the problem. Next, try to determine whether

to call **SBS 1-800-ASK-4SBS** (current Software Maintenance Service contracts) or . When calling for assistance, you will be asked for the Serial Number of your *stroke* system. This number is found in several places: On the first page of the **Disk 1**, and on the *stroke* logo screen.

If you are experiencing random errors with your system, please review the section on Power Conditioning at the beginning of this Appendix.

Error Messages

*Any time you get an error message, you should write down all the information shown on the screen and a note about what you were doing when the error occurred. Then call **SBS** at (800) ASK-4SBS as soon as possible.*

Internal Error: This error message is usually the result of running the **Keystroke** program with insufficient available conventional memory. In many cases, certain modules will run without problems, while others, which are larger and require more memory, will cause the system to crash. You will also get this error if you “shell out” of **Keystroke** using the DOS Shell command on the Utility menu and then load **Keystroke** again (by typing **KEYSTROK**) instead of going back to the one currently running (by typing **EXIT**).

Solution: If an Out of Memory error is encountered, you should first attempt to free up additional base memory. It is recommended that you have at least 550K (532,480 bytes) of conventional memory available (The maximum is 640K on any PC Compatible computer). To check the amount of memory available, type the command **MEM** at the DOS prompt. The amount of memory available will be displayed as Largest Executable Program Size . If you can't free up more memory, you may want to try the **/LOWMEM** Switch (see System Startup Switches).

Printer Not Ready: This error will result when sending data to a printer (also a cash drawer or other external device) that is not ready to receive data.

Solution: First, check to see that the printer (or other device) is on line (usually indicated by a green light on the front of printer). If the printer is not on line, check to see that it is properly plugged in, the power is turned on, and paper is loaded; sometimes it is necessary to disconnect and reconnect the printer cable to ensure a good connection. Check that the printer cable is connected to the same port that is specified in the Printer or Cash Drawer Setup. If you're not sure, try switching either the cable connection or the specified port.

WARNING: Disk is almost full!: This warning message will appear when the **Keystroke** program is started if there is *less than* one megabyte of unused disk space available on the hard disk.

Solution: Free up additional space on the hard disk by removing unnecessary files as soon as possible. If the hard disk becomes completely full, the system will not be able to store additional data and other errors will occur. Refer to the Managing Transaction Data Files section of this Appendix. On newer machines with hard disks greater than 2GB, the **/NOD** switch can be used if you are certain there is ample disk space. See the section on Startup Switches earlier in this Appendix.

Printing Odd Characters Instead of Lines on Reports and Forms:

Solution: Change the character set or symbol set on your printer to one which includes the linedraw characters (single and double lines and intersections; usually called the ASCII Extended or PC-8 character set). This is typically controlled by either a dip switch setting or menu functions on the printer. This problem can also be resolved by not printing documents which use the linedraw characters. Try using an alternate Form File (i.e., one that uses the dash and equal signs instead of unbroken linedraw characters), or changing the Single and Double Line codes in the Printer Driver to 45 and 61 respectively. These are the decimal form of ASCII codes which represent the dash and equal symbols.

Error 5 – Illegal Function Call: This error can be caused by many different things and is usually a sign of some form of corruption (a form file, or the program itself). Back up and reinstall the program disks.

If encountered during installation, indicates an invalid COMMAND.COM file; add **SET COMSPEC=C:\COMMAND.COM** to AUTOEXEC.BAT file.

If occurs on a database record, delete SCRNN#.DAT; if occurs in Label Manager, delete LBL#.DAT; if occurs in Report Manager, delete RPT#.DAT; if occurs in Importer, may have double quotes in source file fields.

Error 6 – Overflow (number too large) Entered too large of a number. It is most often seen when running a report. This is also a sign of data corruption. Try to pinpoint which record (Invoice, Inventory item, etc.) is causing the error and look at the data (any fields with “***”). If incorrect, delete the record, pack, and then reenter it.

Error 7 – Out of Memory: This error is usually the result of running the **Keystroke** program with insufficient available conventional memory (under 640K). **Keystroke** requires 550 – 600K of conventional memory to run properly. Use a utility like MEMMAKER to free up more conventional memory, or try starting the program using the /LOWMEM switch.

Error 9 – Subscript Out of Range: If occurs on the Logo screen, a database index is corrupt. Use /LOWMEM to get into program, then Pack the suspected database. May be a corrupt configuration file; try renaming KSCNF.DAT. If occurs accessing a database screen, may be corrupt screen file (SCRNN#.DAT). May also be memory related.

Error 13 – Type Mismatch: Usually incorrect Import setup or dirty source data. May also be corrupt database screen file; delete SCRNN#.DAT.

Error 14 – Out of String Space: Will sometimes occur when several functions are called upon simultaneously (i.e., stacking several levels of dialog boxes on top of one another). If occurs on a database record, delete SCRNN#.DAT; if occurs in Label Manager, delete LBL#.DAT; if occurs in Report Manager, delete RPT#.DAT, or turn off Detailed function. May also be memory related.

Note: The “#” in the above file names indicates a specific file name/number. Call SBS for assistance in identifying these file names.

Error 24/25/27 – Device Timeout / Device Fault / Out of Paper: Hardware device (printer, drawer, display, scale, etc.) not attached, not ready, or not responding correctly. Check to see that it is turned on, the cables are connected, and that it is on line and has plenty of paper. Make sure you are sending data to the correct port. May be print buffer overflow; slow down baud rate or change COM parameters, if possible.

Error 52 – Bad File Number: This error almost always is a follow-up error to a previous problem. This error by itself does not really help much in figuring out what went wrong. If you get this error, you should first try to remember what you were doing when it occurred, as well as any other errors or warning messages that may have preceded it.

Error 53 – File Not Found: This error can be caused by a missing program or form file; reinstall program disks. It also occurs if you try to run **Keystroke** when you are not in the **Keystroke** program directory, have the **Keystroke** directory in your path statement, or are logged into a network without full read/write access.

Error 55 – File Already Open: Usually encountered when using a cash drawer or pole display. Make sure Keep Port Open setting is OFF.

Error 57 – Device I/O Error: This error is most likely due to a problem with hardware (typically hard drive, network card, or network). If it is the network connection, first try resetting this computer and then possibly the server. Secondly, check your network cards and cables. Your hard disk may also be developing a bad spot; run a hard disk diagnostic program such as ScanDisk or Norton Disk Doctor.

Error 58 – File Already Exists: If encountered during the Pack function, **Keystroke** may be running on another machine. Check file attributes using the ATTRIB command; may be set to read-only.

Error 59 – Bad Record Length: File could be corrupted.

Error 61 – Disk Full: The disk you are writing to has run out of space (less than 1MB available). You must delete some files before continuing (see Managing Disk Space).

Error 62 – Input Past End: This is usually caused by an incorrect or corrupted Form file. Try switching to a different one. Also caused by improper Import source data.

Error 63 – Bad Record Number: This is a sign of a corrupted data file. Back up immediately and then pack the indicated file. Be careful to not back up this corrupted data over the top of an existing (good) backup set. You may need to use the /LOWMEM startup switch to get into **Keystroke** to run the Pack function.

Error 64 – Bad File Name: Usually indicates an incorrect configuration, file name, or setup.

Error 67 – Too Many Files: Increase the value of the FILES= line in your CONFIG.SYS file, and reboot the machine. A number around 80 should be adequate.

Error 68 – Device Unavailable: The program is attempting to send data to a port that does not exist (e.g. a printer is set up on serial port COM2: when there is no COM2: available on the machine). Sometimes occurs when workstation has lost the network connection. The error message usually displays the device or file that is unavailable.

Error 70 – Permission Denied: Network connection lost or sharing disabled. If encountered during Pack function, another machine may be in **Keystroke**.

Error 71 – Disk Not Ready: Retry, check cables, check network connection, reboot if necessary.

Error 75 – Path/File Access Error: Usually due to a lost connection to the network. Check file attributes using the ATTRIB command; may be set to read-only.

Error 76 – Path Not Found: May have a SET TEMP line in your AUTOEXEC.BAT which points to a directory that does not exist. Could also be due to a lost connection to the network.

Warning Messages

#200 – Not enough disk space.

#201 – The actual size of the file is different than the counter that stores the number of records that should be in the file. You usually want to adjust based on size, and pack the suspected file (e.g., File#6 = Transaction file).

#202 – Same as #201.

#203 – File size is >8Mb (64K in version 4.0). The program will display the problem file name in the warning box.

#204 – Free Disk Space <1,048,575 = 1Mb. Hard drives over 2GB will read incorrectly because of the jump from 32 bit to 64 bit. Partition hard drive to less than 2GB or use the /NOD startup switch if you are certain there is enough disk space.

#205 – (AR) Customer Number on the credit memo is not the same as the Customer Number on the Index.

#206 – (AR – Credit Invoice) Program was unable to locate the Invoice in order to apply payments to it. AR will be out of balance. Clear/delete the transaction and re-enter it.

#207 – (AR – Credit Invoice) Program was unable to locate payment. Clear/delete the transaction and re-enter it.

#208 – (AR – UnCredit Invoice) Program was unable to locate the Invoice in order to remove payments from it. AR will be out of balance. Clear/delete the transaction and re-enter it.







#209 – (AR – UnCredit Invoice) Program was unable to locate payments in order to remove it from the Invoice. AR will be out of balance if the Invoice is ever re-saved. Clear/delete the transaction and re-enter it.

#210 – **#211** – (Posting Variance) Err#210 indicates the Stock Number was not found when reading in Variance. Err#211 indicates the Variance would have posted to the wrong item. Call **SBS**.

#212 – (Updating PO from Receiving Function) Item received (which was found in Open Order file) was found on its actual Purchase Order/Invoice. This is caused by a corrupted transaction or Back/Open order file. Rebuild Back/Open order file using the Update function in Delete Utility.

Index

Symbols

- 4.0 Conversion – CONV40.EXE 91
- \$D 104
- \$T 104
-  – Paste Function 26
-  – Copy Function 26
-  – Copy Scripts 14
-  – Select & Auto Select Function 36
-  – Cut 14
-  – Paste 14

A

- Add Item from List Script 16, 18–30
- Add Item from Range Script 16, 18–30
- Add-on Items. *See* Script Editor
- Advanced Features Appendix 39–109
- Alignment Code
 - Label Form File Structure 76
- Alternate Items. *See* Script Editor
- ARPAFF.EXE. *See* Pay Off AR Balances – ARPAFF.EXE
- ASCII Delimited Data Files 78
- Audit Records Function
 - Delete Utility 99
- Auto Produce Function
 - Production 33, 36
- Auto Select Function
 - Production 33, 36
- Auto-Production from DOS Prompt – PROD.EXE 90

B

- Back Orders Function
 - Delete Utility 99
- BackgroundColor 104
- Backing Up 39–40
- Battery Backup 41
- Bike-a-Log On Disk Link – BODLINK.EXE 99
- BODLINK.EXE. *See* Bike-a-Log On Disk Link – BODLINK.EXE

C

- Case Breakdowns. *See* Production: Types of Productions: Reverse Productions
- ChangeNums – CHGNUMS.EXE 101
- Check Header Information on Transaction Files – HEADER.EXE 103
- Check Inventory Script Flag – CHKSCRIPT.EXE 102

- Check Stamp – CHKSTAMP.EXE 102
- CHGNUMS.EXE. *See* ChangeNums – CHGNUMS.EXE
- CHKSCRIPT.EXE. *See* Check Inventory Script Flag – CHKSCRIPT.EXE
- CHKSTAMP.EXE. *See* Check Stamp – CHKSTAMP.EXE
- Clear Function
 - Production 33
 - Script Editor 24
- Command Line Switches. *See* Startup Switches
- Comment. *See* Script Parameters: Comment
- Complete Function
 - Production 33
- Component. *See* Script Parameters: Component
- Configuration Data Files 58–59
- CONV40.EXE. *See* 4.0 Conversion – CONV40.EXE
- Copy Function
 - Script Editor 24, 26
- Copy Next Number – COPYNUM.EXE 103
- COPYNUM.EXE. *See* Copy Next Number – COPYNUM.EXE
- Cost Adjustments 35
- Credit Memos Function
 - Delete Utility 99, 100
- Current Transaction File 61
- Custom Module Data Files 67–80
- Custom Modules
 - Accessing 2–4
 - GL-Link 1, 3–11
 - Importer 1, 28–32
 - Installing 2–4
 - Keystroke** EDC 2
 - MultiStore 2
 - Production 1, 33
 - Script Editor 1, 12–27
- Custom Modules Appendix 1–2
- Customer Display Control Files 70–73
 - Display Strings 73
 - Field ID 72
 - Line Item Message 72
 - Other Transaction Messages (Totals) 72
 - Payment Message 72
- Customer Label Field ID's 75, 76
- Customers Function
 - Importer 29, 31

Click on the Colored Text to Go to That Page

D

Data Field Specifications. *See* Importer: Data Field Specifications
Data Files [57–68](#)
 Configuration Data Files [58–59](#)
 Custom Module Data Files [68–80](#)
 Database Files [59–60](#)
 Database Screen Files [65](#)
 Miscellaneous Data Files [67–80](#)
 Transaction Data Files [60–65](#)
Database Files [59–60](#)
Database Screen Files [65–66](#)
Date Function
 Production [33](#)
Delete Function
 Production [33](#)
 Script Editor [24, 25](#)
Delete Menu
 Delete Utility [99–100](#)
Delete Utility – DELUTIL.EXE [99–100](#)
 Delete Menu [99–100](#)
 Update Menu [99–100](#)
DELUTIL.EXE. *See* Delete Utility – DELUTIL.EXE
Direct Access from DOS Prompt [79–90](#)

E

Edit Function
 Production [33](#)
 Script Editor [24](#)
Error Messages [106–108](#)
Escape Sequences. *See* Printer Drivers
Export from DOS Prompt – KSEXPORT.EXE [82–85](#)
Export Function
 Script Editor [24, 26](#)
Extracting Data From **Keystroke** [78–79](#)

F

Fail-Safe Network [41–58](#)
Field Parameters
 Label Form Files [74–76](#)
File Management [53–68](#)
File Menu
 Production [33, 38](#)
 Pack Productions Function [33, 38](#)
 Pack Templates Function [33, 38](#)
File Split [61](#)
Fixed Quantity. *See* Script Parameters: Fixed Quantity
ForegroundColor [104](#)
FORM2LBL.EXE. *See* Print Labels from Invoice Form – FORM2LBL.EXE
Format Qty Column on Invoice Form – FORMQTY.EXE [97](#)
FORMQTY.EXE. *See* Format Qty Column on Invoice Form – FORMQTY.EXE

G

GL-Link
 Linking with Specific Programs [8–11](#)
 Menus [3](#)
 Setup Menu [3, 4–8](#)
GL-Link from DOS Prompt – GLLINK.EXE [85, 85–86](#)
 Optional Switches [86](#)
 Other Switches [86](#)
GL-Link Module [1, 3–11](#)
GLLINK.EXE. *See* GL-Link from DOS Prompt – GLLINK.EXE

H

Header Parameters
 Label Form Files [74](#)
HEADER.EXE. *See* Check Header Information on Transaction Files – HEADER.EXE

I

IMPEXE. *See* Importer from DOS Prompt – IMPEXE
IMPALTIC.EXE. *See* Import Alternate Inventory Codes – IMPALTIC.EXE
Import Accounts Receivable Balances – IMPORTAR.EXE [97–98](#)
 Optional Switches [97](#)
Import Alternate Inventory Codes – IMPALTIC.EXE [98–99](#)
 Optional Switches [98](#)
Import Function
 Script Editor [24, 27](#)
Import Menu
 Importer [29, 31](#)
 Customers Function [29](#)
 Inventory Function [29, 31](#)
 Vendors Function [29, 31](#)
IMPORTAR.EXE. *See* Import Account Receivable Balances – IMPORTAR.EXE
Importer from DOS Prompt – IMPEXE [86](#)
 Optional Switches [86](#)
 Other Switches [86](#)
Importer Module [1, 28–32](#)
 Data Field Specifications [32–38](#)
Inventory Departments Function
 GL-Link [3, 7](#)
Inventory Function
 Importer [29, 31](#)
Item List [19](#)
Item Range [19](#)

Click on the Colored Text to Go to That Page

K

- Keeping Networked Clocks Synchronized [62](#)
- Keystroke** EDC [2](#)
- Kitting. *See* Script Editor
- KSEXPORT.EXE. *See* Export from DOS Prompt –
KSEXPORT.EXE
- KSSCHDLR.EXE. *See* Program Scheduler –
KSSCHDLR.EXE

L

- Label Form File Structure [73–77](#)
 - Field Parameters [74](#)
 - Field ID [74](#)
 - Text Field [74](#)
 - Header Parameters [74](#)
 - Control Flag [74](#)
 - Description Field [74](#)
 - Label Width [74](#)
 - Number Across [74](#)
 - Number of Lines [74](#)
- Large Numbers – LRGPRINT.TXT [103–104](#)
- Line Number
 - Label Form File Structure [76](#)
- Link Menu
 - GL-Link [3, 8](#)
 - Sales, AR Function [3, 8](#)
 - Transfers Function [3, 11](#)
- Link To Function
 - GL-Link [3, 4](#)
- LRGPRINT.TXT. *See* Large Numbers –
LRGPRINT.TXT

M

- Managing Disk Space [68](#)
- Max Items. *See* Script Parameters: Max Items
- Menus
 - Delete Menu
 - Delete Utility [99–100](#)
 - File Menu
 - Production [33, 38](#)
 - GL-Link [3](#)
 - Link Menu [3, 8](#)
 - Setup Menu [3, 4–8](#)
 - Import Menu
 - Importer [29, 31](#)
 - Importer
 - Import Menu [29, 31](#)
 - Setup Menu [29](#)
 - Link Menu
 - GL-Link [3, 8](#)
 - Production
 - File Menu [33, 38](#)
 - Production Menu [33, 36](#)
 - Reports Menu [33, 37](#)
 - Template Menu [33, 36](#)

- Production Menu
 - Production [33, 36](#)
- Reports Menu
 - Production [33, 37](#)
 - Script Editor [24, 25](#)
- Script Editor
 - Reports Menu [24, 25](#)
 - Script Menu [24](#)
 - Special Menu [24, 26](#)
- Script Menu
 - Script Editor [24](#)
- Setup Menu
 - Importer [29](#)
- Special Menu
 - Script Editor [24, 26](#)
- Template Menu
 - Production [33, 36](#)
- Update Menu
 - Delete Utility [99–100](#)
- Message. *See* Script Parameters: Message
- Min Items. *See* Script Parameters: Min Items
- Miscellaneous Data Files [67](#)
- Miscellaneous Files [103–104](#)
 - Large Numbers – LRGPRINT.TXT [103–104](#)
 - Screen Saver – SCRNSAVE.TXT [104](#)
- Multi-Drawer [46–56](#)
- MultiStore [2](#)

N

- Network Version Features [44–46](#)
- NUMBERS.EXE. *See* Print Unused Stock Numbers –
NUMBERS.EXE

O

- Open Orders Function
 - Delete Utility [99, 100](#)
- Out of Stock Only. *See* Script Parameters: Out of Stock
Only

P

- Pack Function
 - Script Editor [24, 26](#)
- Pack Productions
 - Production [33, 38](#)
- Pack Templates Function
 - Production [33, 38](#)
- Parameters Function
 - GL-Link [3, 5](#)
- Parts Assembly. *See* Script Editor
- Paste Function
 - Script Editor
 - Special Menu [24, 26](#)
- Pay Off AR Balances – ARPAYOFF.EXE [80](#)
- Payment Types Function
 - GL-Link [3, 8](#)

Click on the Colored Text to Go to That Page

Payments Paid Function
Delete Utility 99

Payments Received Function
Delete Utility 99

Post Dated Transactions 61–62, 62

Posting Accounts Function
GL-Link 3, 6

Power Conditioning 41

Price Tag Field ID's 75

Print Code
Label Form File Structure 76

Print Labels from Invoice Form – FORM2LBL.EXE
95–96

Print Unused Stock Numbers – NUMBERS.EXE 102

Printer Drivers 68–70, 69–70

PROD.EXE. *See* Auto-Production from DOS Prompt –
PROD.EXE

Production Menu
Production 33, 36
Clear Function 33
Complete Function 33
Date Function 33
Delete Function 33
Edit Function 33
Recall Function 33

Production Module 1, 33–38
Types of Productions 35–38

Productions from Components 35

Productions Function
Production 33, 37

Program Scheduler – KSSCHDLR.EXE 92

Prompt for Comment Script 16, 16–17

Prompt For Quantity. *See* Script Parameters: Prompt
For Quantity

Q

Qty Allocated/On Layaway Function 100

R

Recall Function
Production 33
Script Editor 24, 25

RECMRG.EXE. *See* Recurring Invoices/Recovery
Merge – RECMRG.EXE

Recovery Merge – RECMRG.EXE 92

Recurring Invoices/Recovery Merge – RECMRG.EXE
92–94

Register Numbers 45–53

REMDUPES.EXE. *See* Remove Duplicate Sales
Transactions – REMDUPES.EXE

Reminder Message Script 16

Remove Duplicate Sales Transactions –
REMDUPES.EXE 103

Repeat Items. *See* Script Parameters: Repeat Items

Replace Scripted Item. *See* Script Parameters: Replace
Scripted Item

Replacement Items. *See* Script Editor

Report Cues – RPT.EXE 80–82
Optional Switches 81
Other Switches 82

Reports Menu
Production 33, 37
Productions Function 33, 37
Templates Function 33, 37
Script Editor 24, 25
Scripts Function 24, 25

Resource Files 55–57

Reverse Productions 35

ROA/Invoices Function
Delete Utility 100

Roll Up Cost. *See* Script Parameters: Roll Up Cost
RPT.EXE. *See* Report Cues – RPT.EXE

S

Sales, AR Function
GL-Link 3, 8

Save Function
Script Editor 24, 25

Screen Saver – SCRNSAVE.TXT 104

Script Editor
Adding Scripts 14
Copying Scripts 14
Editing Scripts 15
Managing Script Data Files 27
Saving Scripted Items 15
Saving Scripts 15–16
Setting Up Scripted Items 13–27
Testing Scripts 15
Types of Scripts 16–29

Script Editor Module 1, 12–27

Script Menu
Script Editor 24
Clear Function 24
Delete Function 24, 25
Edit Function 24
Recall Function 24, 25
Save Function 24, 25

Script Parameters 20–29
Comment 20
Component 21
Fixed Quantity 23
Max Items 23
Message 20
Min Items 23
Out of Stock Only 22
Prompt For Quantity 24
Quantity 22
Repeat Items 22
Replace Scripted Item 21
Roll Up Cost 22

Scripts Function
Script Editor 24, 25

SCRNSAVE.TXT. *See* Screen Saver –
SCRNSAVE.TXT

Click on the Colored Text to Go to That Page

- Select Function
 - Production [33, 36](#)
 - Setting Up Report Cues for DOS Commands [81](#)
 - Setup – SETUP.EXE [91](#)
 - Setup Menu
 - GL-Link [3, 4–8](#)
 - Inventory Departments Function [3, 7](#)
 - Link To Function [3, 4](#)
 - Parameters Function [3, 5](#)
 - Payment Types Function [3, 8](#)
 - Posting Accounts Function [3, 6](#)
 - Importer [29](#)
 - Customers Function [29, 31](#)
 - Inventory Function [29, 31](#)
 - Vendors Function [29, 31](#)
 - SETUP.EXE. *See* Setup – SETUP.EXE
 - Show Document – SHOWDOC.EXE [90](#)
 - SHOWDOC.EXE. *See* Show Document – SHOWDOC.EXE
 - Source Data [28](#)
 - Special Menu
 - Copy Function [24, 26](#)
 - Script Editor [24, 26](#)
 - Export Function [24, 26](#)
 - Import Function [24, 27](#)
 - Pack Function [24, 26](#)
 - Paste Function [24, 26](#)
 - Splitting Transaction Files [61](#)
 - Stamp – STAMP.EXE [102](#)
 - STAMP.EXE. *See* Stamp – STAMP.EXE
 - Stand-Alone Mode [42–43](#)
 - Stand-Alone Utilities [90–103](#)
 - 4.0 Conversion – CONV40.EXE [91](#)
 - Bike-a-Log On Disk Link – BODLINK.EXE [99](#)
 - ChangeNums – CHGNUMS.EXE [101](#)
 - Check Header Information on Transaction Files – HE [103](#)
 - Check Stamp – CHKSTAMP.EXE [102](#)
 - Copy Next Number – COPYNUM.EXE [103](#)
 - Delete Utility – DELUTIL.EXE [99](#)
 - Format Qty Column on Invoice Form – FORMQTY.EXE [97](#)
 - Import Accounts Receivable Balances – IMPORTAR.EXE [97](#)
 - Import Alternate Inventory Codes – IMPALTIC.EXE [98](#)
 - Print Labels from Invoice Form – FORM2LBL.EXE [95–96](#)
 - Print Unused Stock Numbers – NUMBERS.EXE [102](#)
 - Program Scheduler – KSSCHDLR.EXE [92](#)
 - Recurring Invoices/Recovery Merge – RECMRG.EXE [92](#)
 - Remove Duplicate Sales Transactions – REMDUPES.EXE [103](#)
 - Setup – SETUP.EXE [91](#)
 - Show Document – SHOWDOC.EXE [90](#)
 - Stamp – STAMP.EXE [102](#)
 - Startup Switches [47, 47–53](#)
 - /B – Black & White [50](#)
 - /C – Configuration [49](#)
 - /D – Data Directory [47](#)
 - /E – Data Extension [48](#)
 - /F – Form Files [49](#)
 - /G – Go Direct to Module [52](#)
 - /H, /? – Help [47](#)
 - /K – **Keystroke** Resource Directory [51](#)
 - /LOWMEM – Low Memory [50](#)
 - /NEWFILE – Split Transaction File [52](#)
 - /NOD – No Disk Space (or File Size) Check [53](#)
 - /NOSNOW – No Snow [51](#)
 - /NOSPLIT [51](#)
 - /P – Printers [50](#)
 - /R – Register Number [48](#)
 - /S – Sound Off [50](#)
 - /TIME – Date/Time Prompt [52](#)
 - /TUTOR – Tutorial Data [48](#)
 - /U – User Temporary Directory [51](#)
 - XTKB and ATKB – Keyboard Type [51](#)
 - StoreLink Inventory Update – STRUPDTE.EXE [90](#)
 - STRUPDTE.EXE. *See* StoreLink Inventory Update – STRUPDTE.EXE
 - Sub-assembly. *See* Script Editor
 - Subdirectories [53–55](#)
 - Data Directory – DATA\ [54](#)
 - Documents Directory – DOC\ [55](#)
 - EDC Directory – EDC\ [55](#)
 - Files Directory – FILES\ [55](#)
 - Forms Directory – FORMS\ [54](#)
 - Help Directory – HELP\ [54](#)
 - Installation Directory – TEMP\ [55](#)
 - Old Summary Files – DATA\SUMMARY\OLD\ [54](#)
 - Reports Directory – REPORTS\ [55](#)
 - Summary Files – DATA\SUMMARY\ [54](#)
 - Tutorial Directory – TUTOR\ [54](#)
 - Substitute Items. *See* Script Editor
 - Suggested Items. *See* Script Editor
 - Summarize History – SUMMARY.EXE [87–89](#)
 - Startup Switches [88](#)
 - Summary Data Files [66, 89](#)
 - SUMMARY.EXE. *See* Summarize History – SUMMARY.EXE
 - Superseded Items. *See* Script Editor
- ## T
- Tag-along Item Script [16, 17–18](#)
 - Tag-along Items. *See* Script Editor
 - Template Menu
 - Production [33, 36](#)
 - Auto Produce Function [33, 36](#)
 - Auto Select Function [33, 36](#)
 - Delete Function [33](#)
 - Recall Function [33](#)
 - Select Function [33, 36](#)

Click on the Colored Text to Go to That Page

Templates Function
 Production [33, 37](#)
Temporary and Resource Files [55–57](#)
 Keystroke Resource Files – /K [55](#)
 Temporary File Names [56](#)
 Temporary User Files – /U [56](#)
TimeInterval [104](#)
Token Definitions [95](#)
Tokens [95](#)
Transaction File Splits Function
 Delete Utility [100–103](#)
Transaction File Storage Capacity [64](#)
Transfers Function
 GL-Link [3, 11](#)

U

Update Customers Function [100](#)
Update Menu
 Delete Utility [99–100](#)
Updating **Keystroke** Data Using Other Programs [79–80](#)
Using **Keystroke** Data With Other Programs [78–79](#)
 ASCII Delimited Data Files [78](#)
 Extracting Data From **Keystroke** [78](#)
 Updating **Keystroke** Data Using Other Programs [79](#)

V

Vendor Label Field ID's [75, 76](#)
Vendors Function
 Importer [29, 31](#)

W

Warning Messages [109](#)

X

xPos [104](#)

Y

yPos [104](#)