

# Keystroke™

*Point Of Sale*

*MultiStore Utilities*



Specialized Business Solutions

# ***Specialized Business Solutions License Agreement***

**IMPORTANT! Please read carefully:** Before installing or using any software published by **Specialized Business Solutions**, you should read and understand the following license agreement. By using **Keystroke** or any other software published by **Specialized Business Solutions**, a California corporation, you are agreeing to be bound by the terms of this agreement.

**Specialized Business Solutions** (“SBS”) grants you the right to install one copy of the enclosed software on a single computer or local area network (LAN) and to use this copy of the software in the operation of a single business entity. A “business entity” is defined as a place of doing business (such as a storefront, office or warehouse), or a unique commercial business differentiated by advertising, signage or business name which can be perceived by the public as a separate business. A separate licensed copy of **Keystroke** with a unique serial number must be purchased for each separate business entity using **Keystroke**.

The **Keystroke** software is owned by **SBS** and is protected by United States and international copyright laws. Therefore, no part of this software or accompanying documents may be reproduced or transmitted in any form or by any means (electronic, mechanical, magnetic, or otherwise) without the written permission of **SBS**. The purchaser may transfer one copy of the software to a single hard disk provided the original copy is used solely for backup purposes.

**SBS** makes no warranties implied or inferred with respect to this software, the accompanying documents, and any accompanying hardware. Further, **SBS** reserves the right to change the information in this software or accompanying documents without notice. **SBS** shall not be liable for errors contained herein or for any damages whatsoever (including, without limitation, damages for loss of business profits, business interruption, loss of business information, or other pecuniary loss), resulting from the use of or inability to use this product. Any and all disputes arising out of or related in any way to this license agreement shall be resolved through arbitration procedures, as recommended by the American Arbitration Association, in Long Beach, California.

Copyright © by Specialized Business Solutions, 1988-2002. All rights reserved.  
**Keystroke** is a trademark of Specialized Business Solutions.  
All other products mentioned are trademarks or registered trademarks of their respective companies.



## **Specialized Business Solutions**

<i>Service</i>	<i>R &amp; 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 Service: 800•275•4727*  
*Non-Emergency Support: 970•468•4842*

# Table of Contents

<b>MultiStore Utilities</b> .....	<b>5</b>
<b>Overview</b> .....	<b>7</b>
<b>Data Configurations</b> .....	<b>9</b>
<b>Managing Data Files</b> .....	<b>11</b>
<b>Managing Transaction Data</b> .....	<b>12</b>
<b>Managing Inventory Data</b> .....	<b>17</b>
<b>Managing Store Data</b> .....	<b>18</b>
<b>StoreLink Menus</b> .....	<b>20</b>
<b>Transfer Menu</b> .....	<b>20</b>
Clear Function .....	20
Edit/Print Function .....	21
Copy Function .....	21
Delete Function .....	21
Recall Function .....	22
Number/Date Function .....	22
Print/Save Function .....	22
Parameters Function .....	22
<b>Store Menu</b> .....	<b>26</b>
Current Function .....	26
Transfer To Function .....	27
Setup Function .....	27
Update Inventory Function .....	28
<b>Reports Menu</b> .....	<b>29</b>
Inventory Function .....	29
Transfers Function .....	29
<b>Special Menu</b> .....	<b>30</b>
Import Line Items Function .....	30
Pack Function .....	30

<b>Miscellaneous/Error Checking Utilities.....</b>	<b>31</b>
<i>ChangeNums .....</i>	31
<i>CheckNum .....</i>	31
<i>CheckScript .....</i>	32
<i>CopyClock .....</i>	32
<i>CopyNum .....</i>	32
<i>DateName .....</i>	33
<i>FindFile .....</i>	33
<i>GetMail .....</i>	34
<i>Keystroke Polling .....</i>	35
<i>KSScheduler .....</i>	35
<i>Remove Duplicates .....</i>	35
<i>SendMail .....</i>	36
<i>Stamp/CheckStamp .....</i>	36
<i>Store Update .....</i>	37
<b>Setting up MultiStore .....</b>	<b>38</b>
<b>Sample Polling Procedures .....</b>	<b>42</b>
<i>Sample Polling Script with 2 Stores, Both Live .....</i>	43
<i>Sample Polling Script with a Remote Store .....</i>	45
<i>Sample pcAnywhere Script Files .....</i>	48
<b>MultiStore Polling via Email .....</b>	<b>50</b>
<b>MultiStore Polling via FTP .....</b>	<b>58</b>
<b>Troubleshooting .....</b>	<b>64</b>
<b>Variations and Restrictions .....</b>	<b>66</b>
<b>Using GL- Link with MultiStore .....</b>	<b>67</b>
<b>Index.....</b>	<b>71</b>

# *MultiStore Appendix*

MultiStore is a set of software utilities which can be used in conjunction with the **Keystroke POS** system to help you manage sales and inventory in multiple stores and/or warehouses. The MultiStore utilities include a variety of stand-alone programs that, when properly integrated into automated polling scripts and/or batch programs, allow various levels and methods of data consolidation depending on the needs of your business. Some of the most common benefits of the MultiStore utilities include:

- Nightly polling of data from store locations to a company headquarters. Transactions, Customers, and Inventory databases are available for review/maintenance by management at company headquarters. (Extract, StoreMerge, and StoreLink/Update Inventory utilities)
- Sales and purchase transactions from all locations can be consolidated into one set of **Keystroke** data files to represent company-wide history and status. (Extract and StoreMerge utilities)
- Accounts Receivable can be consolidated between several stores so a Customer can shop at any store and receive a single statement.
- Company-wide financial data can be consolidated to a common General Ledger. (Extract, StoreMerge, and GL-Link; not a part of MultiStore)
- From any store location, look up Quantity On Hand (QOH) of Inventory items located in other stores and/or warehouses. (Inventory Link utility)
- Consolidated Inventory level reports including QOH per store and totals. (StoreLink utility)

Each of the above features can be implemented individually or collectively, depending upon the needs of your business. In addition, many of the MultiStore utilities can be configured in a variety of ways, providing several levels of integration. In this documentation, we will describe the most common methods of configuring each of the MultiStore utilities.

## ***Software Licensing Reminder***

In accordance with the **SBS License Agreement** (see inside the front cover of the main User Guide), a separate **Keystroke** package with a unique Serial Number must be purchased for each store, warehouse, or other location. This also applies to Custom modules such as GL-Link (with the exception of the MultiStore module, for which a single copy may be licensed for use with a set of Serial Numbers).



A separate store or location is defined as any time the **Keystroke** software is used in any of the following: separate companies, separate buildings, different store names, separate stand-alone computers (including notebook or portable machines used in vehicles for the purpose of sales or delivery), or more than one Local Area Network (LAN) operating system. Running the **Keystroke** software on a Wide Area Network (WAN) does not constitute any exceptions to the **SBS License Agreement**. A separate **Keystroke** package with a unique Serial Number must be purchased for each store, warehouse, or other location at which the **Keystroke** software is to be used, regardless of how the program is run or data is to be transferred.

## MultiStore Utilities

---

**NOTE:** All of the below utilities are explained in greater depth later on in this Appendix.

---

- ChangeNums** (CHGNUMS.EXE) – Can be used to renumber the Customer and/or Inventory databases. This may be required if stores were using **Keystroke** prior to installing MultiStore and therefore possess different data records.
- CheckNum** (CHKNUM.EXE) – Checks to make sure that the two specified database files have the same Next Number stored in them. Used to make sure the customer database was updated correctly before copying back to the remote store.
- CheckScript** (CHKSCRIPT.EXE) – Checks to see if each Inventory item has a Script attached to it, and updates the hidden flag on that item (i.e., it verifies Scripts are available). This should be run if Scripts are copied from one data set to another.
- CheckStamp** (CHKSTAMPEXE) – Checks 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 STAMPEXE program. Used to make sure the complete set of data files were transferred correctly.
- CopyClock** (COPYCLK.EXE) – Copies the TimeClock fields (Last In/Out) from the file KSCLK.TMP to the file KSCLK.DAT. This is used if you copy a Master Clerk database to each store but still want to be able to use the Time Clock function at each location.
- CopyNum** (COPYNUM.EXE) – Copies the Customer Next Number from the merge file KSTRN001.MRG to the Customer data file KSCST.DAT. Allows each store to enter new Customers with unique numbers.
- DateName** (DATENAME.EXE) – Changes the name of a file to a name representing the current date (in YYMMDD format). If a file already exists with that name, the program will append a dash and a letter (e.g., -A, -B, -C, etc.) to the end of the file name. The program will create a file called DATENAME.TXT that will contain the names of the files created. This utility can be used to provide a backup for each day's KSTRN001.MRG file.
- Extract** (EXTRACT.EXE) – Extracts *new* transactions, customers, and/or serial numbers from your current data files into the file KSTRN001.MRG. This file can then be used with the StoreMerge program to merge this new information into another set of data files.
- FindFile** (FINDFILE.EXE) – Finds any files containing the designated text. Used with CHKSTAMPEXE to check for error conditions.
- GetMail** (GETMAIL.EXE) – Reads entries from the Inbox of the specified MAPI server. Entries that match the specified parameters will have their attachments saved to a specified path and will then be deleted. Used in MultiStore polling procedures to receive data.
- InventoryLink** (INVLINK.EXE) – Reads the current Inventory data file and then writes the QOH information to a file (KSIL#.DAT) which can then be read from an Inventory record by pressing   to display the QOH for all stores.

**KS Scheduler** (KSSCHDLR.EXE) – Can be used in conjunction with DOS batch programs to cause other programs to wait until a preset time of day before running.

**RemDupes** (REMDUPES.EXE) – Deletes any duplicate Sales Invoices or ROA payments (should only be used as a last resort).

**SendMail** (SENDMAIL.EXE) – Posts a message (with a file attachment) to the Outbox of a specified MAPI server. Used in MultiStore polling procedures to send data.

**Stamp** (STAMP.EXE) – Stamps the specified **Keystroke** database files with the current Date/Time. Used with CHKSTAMP.EXE to make sure data sets are kept together.

**StoreMerge** (STOREMRG.EXE) – Merges the existing transactions from the merge file KSTRN001.MRG into existing data files.

**StoreUpdate** (STRUPDTE.EXE) – Runs the Update Inventory function from the StoreLink module.

## **MultiStore Overview**

This introductory section is intended to outline the “big picture” of how MultiStore is designed, what it is capable of doing, and the basic requirements such as other hardware and software that will be needed. We’ll also discuss some of the minor implications that you should be aware of before getting started in setting up **Keystroke** to help you manage data for multiple stores and/or warehouses.

The idea of managing data that is constantly changing simultaneously in multiple locations is by no means a simple task. Although there are other ways of achieving this task (i.e., using a Wide Area Network (WAN), where all stores are always connected real-time via internet or phone lines), those methods are often cost prohibitive (i.e., requiring special dedicated phone lines and additional equipment) and can also lead to other difficulties. The **Keystroke** MultiStore utilities offer a less expensive alternative by which the internet or an off-the-shelf communications software package and standard modems can be used to transfer only the data that has changed on a nightly (or other regularly scheduled) basis.

For most businesses, the most useful benefit of MultiStore is the ability to transfer transaction data from remote store locations to a central (“Master”) location (typically either a main store or warehouse). This requires several steps for each location (all automated into the Polling Script). First, new transactions are extracted from the live data files at the remote store. The resulting file is then transferred back to the Master location, and merged into a copy of the data files. The result is that each morning at the Master location there will be an exact copy of each store’s data files, and/or a separate set of data files that represents a consolidation of all stores’ transactions. Along with the transaction data, new Customers are transferred and Inventory levels are updated. The consolidated data files will then have a complete mailing list for all stores, allow reports to be run for the whole enterprise, and provide a single Accounts Receivable statement for customers that shop at more than one store. The procedure for setting up these capabilities (Extract and StoreMerge utilities) is described in the Managing Transaction Data section of this Appendix.

Another popular benefit of MultiStore is the ability to lookup the Quantity On Hand of Inventory items at other locations. This is accomplished by generating a small file called an Inventory Link file (e.g., **KSIL2.DAT**) for each store location. This file contains the current Inventory Quantities for that particular store. These files are then transferred to each of the other locations. You can then press **SHIFT F5** from an Inventory database record or list box to view the quantities at other locations. These procedures can also be integrated into the Polling Scripts that are executed each night, and can be executed more frequently if necessary. The quantities available through this feature are only as accurate as of the last time the InvLink files were created (typically nightly).

The StoreLink module includes a variety of special features which can be used at the Master location for updating Inventory data. This includes transferring Inventory between stores and maintaining a Master Inventory for all store locations, as well as generating reports with QOH for each store with consolidated totals.

Company-wide financial data can be consolidated to a common General Ledger or posted separately by each store using the **Keystroke** GL-Link module. Using the GL-Link module, journal entries can be generated independently for sales at each store running the **Keystroke** program. The GL-Link module can be used at the physical stores, and the resulting files transferred to the Master location, or if data is transferred via Extract/StoreMerge, the GL-Link module can be used at the Master location to generate journal entries from the merged data. The journal entry data can then be posted to either a single General Ledger or you may post to separate sub-accounts in the same General Ledger. Posting to separate sub-accounts is most common.

Now that we've touched on the overall capabilities of the **Keystroke** MultiStore utilities, we'll describe the other hardware and software required to properly set up the MultiStore utilities. Before you get too far along configuring MultiStore for your business, be sure to also review the Variations and Restrictions section at the end of this Appendix.

## Requirements

Although you can configure the **Keystroke** system to run on a Wide Area Network (WAN), the most common method of managing data for multiple locations is using the MultiStore utilities with an off-the-shelf communications software package, standard modems, and regular phone lines. Other options such as using the internet or email are also becoming available and more popular. The procedures outlined below will also work for these new communication methods with little alteration.

---

*NOTE: If you want to consider using a WAN, please contact your local authorized **Keystroke** dealer for assistance. Special hardware, software, and phone lines may be required for WAN's, which are different from those discussed below.*

---

### Communications Software

**Keystroke** does not handle the actual transfer of files; therefore, an off-the-shelf communications software program is required in addition to the MultiStore utilities. The communications software must have the ability to successively execute a set of commands that are integrated into a custom Polling Script (i.e., pseudo-program, similar to a batch program). The communications software should be able to execute the Polling Script unattended at a predefined time of day (usually at night, after all stores are closed for business).

The packages most commonly used with MultiStore are: CloseUp v6.5 for DOS and Windows, published by Norton-Lambert; pcANYWHERE v9.2 for Windows 95/98/NT, by Symantec; Blast Professional for DOS, and Blast Data Pump. Your local **Keystroke** dealer should be able to obtain any of these packages for you. Sample MultiStore Polling Scripts are available for CloseUp, pcANYWHERE, and Blast, and are included with the MultiStore software on the **Keystroke** CD (also available from **Keystroke** Technical Support, or our web site at [www.KeystrokePOS.com](http://www.KeystrokePOS.com)).

### **Modems**

Since the technology behind computer hardware changes so quickly, it is difficult to specify a recommended brand, model, or even speed for modems. We do, however, recommend that you use name brand-modems, and purchase the fastest available designed for use with standard phone lines. Whenever possible, it is also a good idea to use the same type of modem at all locations.

### **Phone Lines**

In most areas, you should be able to use standard telephone lines. ISDN lines are also a nice option, if they are available. Depending on how frequently and at what time of day you plan to transfer data, you may want to consider adding an additional phone line at each location. In some geographical areas (especially remote rural areas where telephone service has not yet been updated with modern technology), communications may be difficult and relatively unreliable. If you believe this to be a potential problem in your area, you may want to research this issue by asking other businesses and/or your local telephone service provider for advice.

## **Data Configurations**

The MultiStore utilities are designed to allow for maximum flexibility in how data will be handled. There are several data configurations that are most typically used; for clarity we've come up with names to differentiate between each method of configuring data: Virtual Master, Main Store Master, and Warehouse Master.

These data configurations are based on Version 4.x of the **Keystroke** program, and may need to be slightly modified if you are running an earlier version of the software.

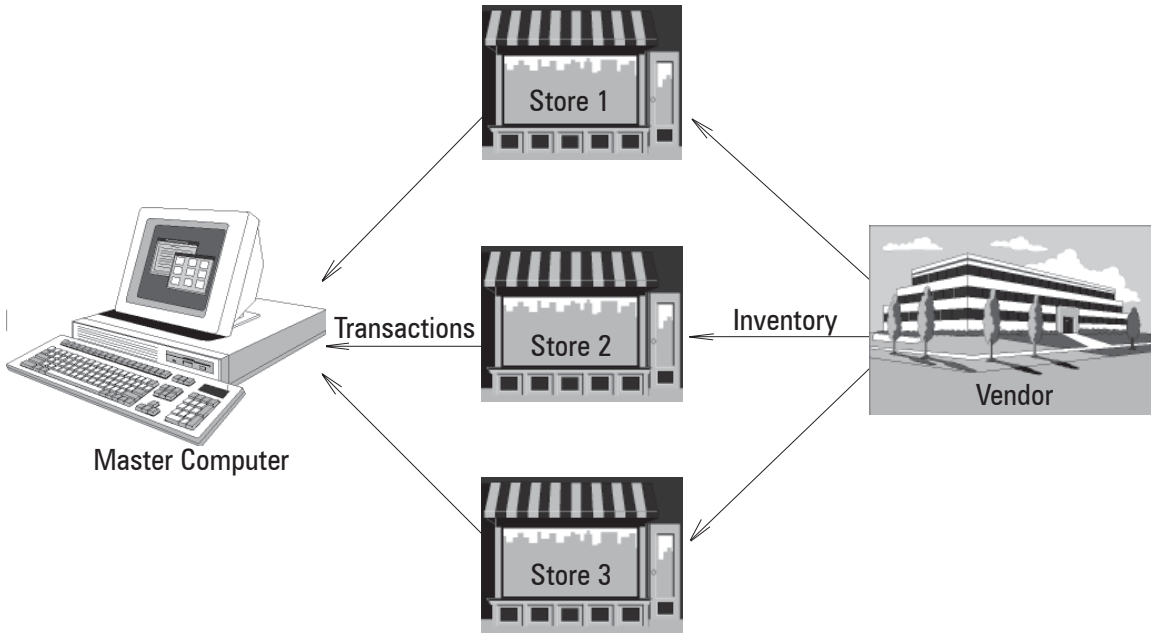
### **Virtual Master**

The Virtual Master configuration is probably the most commonly used method. Businesses using this method typically do purchasing directly from Vendors at each store location. This setup can also be used when ordering is done from a centralized ("Master") location.

With the Virtual Master configuration, the Master data files consist of consolidated transactions (sales and purchases) and Customers (including Accounts Receivable) from all store locations. The Inventory data files are used for centralized maintenance (e.g., adding new items, updating prices, etc.), but are not used for tracking actual quantities, thus the name "Virtual".

On the computer where the Master data files are kept, subdirectories must also be created (normally under the **Keystroke** program directory) for the purpose of storing a copy of each store's data. The Master computer may be located at one of the stores or at another office location. If the Master store is also one of the stores, that store's subdirectory will contain live files for that particular store. A typical setup at the Master store would include the following subdirectories:

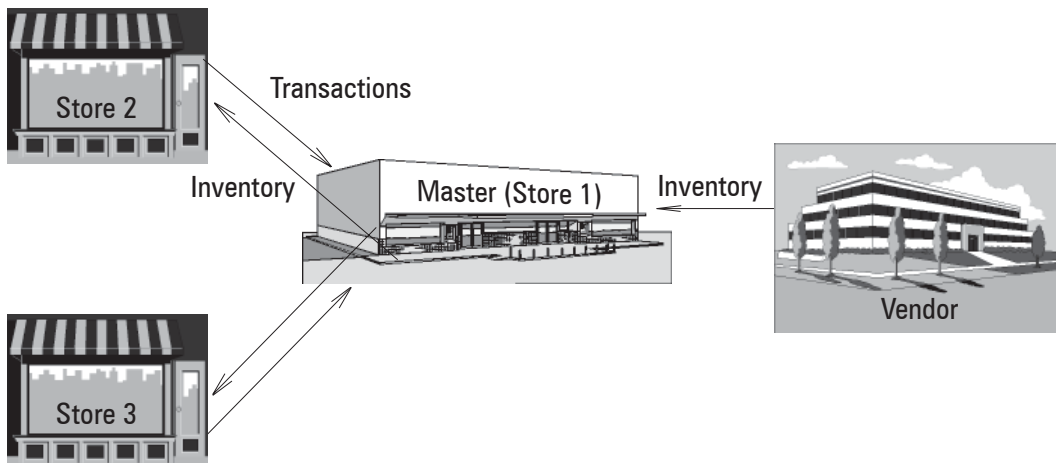
- F:\KEYSTROK\DATA\** – Master data files (Store Number 0)
- F:\KEYSTROK\STORE1\** – Store Number 1 data files
- F:\KEYSTROK\STORE2\** – Store Number 2 data files
- F:\KEYSTROK\STORE3\** – Store Number 3 data files



**Main Store Master**

Using the Main Store Master configuration, the Master Store (which could be a warehouse and/or a retail store) typically purchases and receives all Inventory, then transfers items to the other store locations. Transaction data from stores can be consolidated with the Master Data or kept in separate subdirectories. All Inventory maintenance is done in the Master Store data files (Store 1), which can be used to update Inventory files for each store and/or the consolidated data.

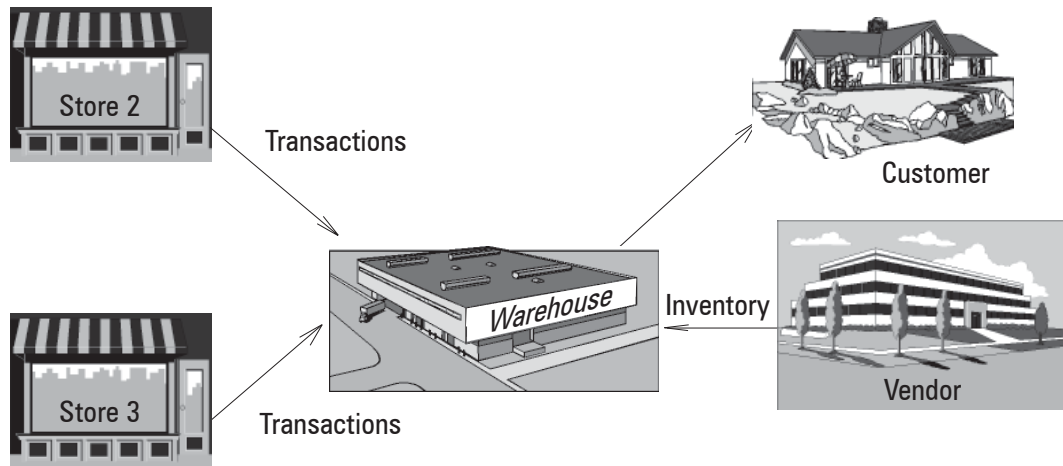
- F:\KEYSTROK\DATA\** – Consolidated data files
- F:\KEYSTROK\STORE1\** – Store Number 1 data files
- F:\KEYSTROK\STORE2\** – Store Number 2 data files
- F:\KEYSTROK\STORE3\** – Store Number 3 data files



## Warehouse Master

The Warehouse Master configuration method is most typically used by furniture stores, where all Inventory (except for display units) is physically kept at a warehouse location. Each store location has no real Inventory quantities; they take orders which are in turn taken from stock at the warehouse and delivered.

**F:** \KEYSTROK\DATA\ – Warehouse data files (Store Number 1)  
**F:** \KEYSTROK\STORE2\ – Store Number 2 data files  
**F:** \KEYSTROK\STORE3\ – Store Number 3 data files



## Managing Data Files

Before using the StoreLink module, it is important that you first understand how the **Keystroke** data files are handled. The information for each store or location is stored in a separate set of data files. Each set of data files should be identified by a unique directory (i.e., **KEYSTROK\DATA\** or **KEYSTROK\STORE1\**). The default (unless otherwise specified) directory and extension is **KEYSTROK\DATA\\*.DAT**, where **KEYSTROK\DATA\** represents the directory where all **Keystroke** data files are located and **.DAT** represents the file extension of the data files stored in that directory. **C:\KEYSTROK\DATA\** is commonly referred to as the DOS path, where **C:** represents the letter assignment of the hard disk. If operating a networked system or LAN, **C:** is often replaced with **F:** or another letter designation.

In order to access files in different directories, you must specify the data directory when running **Keystroke** by entering **KEYSTROK /D**, followed by the desired directory name (e.g., **KEYSTROK /D C:\KEYSTROK\STORE2**).

---

*NOTE: The entire path of the data directory does not need to be specified if it is below the program directory (i.e., the above example would also work by entering **KEYSTROK /D STORE2**).*

---

By keeping track of the *extensions* and *directories* for each store (or set of data files), StoreLink provides a convenient way to tell **Keystroke** which set of data files to use. From here on, we'll refer to each set of data files as a store. By switching from one store to another within the StoreLink module, you can easily edit and generate reports using data from any store without having to exit the **Keystroke** program. However, it is often better (and less confusing) to have a separate icon (or shortcut or menu selection) to run **Keystroke** using the data for each store.

## Managing Transaction Data

Each store or location running the **Keystroke** system generates its own transaction data. Other data such as Inventory and Customer records are also updated as transactions are entered. All of this data can be transferred or copied between locations using a custom procedure (called a “Polling Script”) that automatically executes at a preset time each day (typically at night after business close, or early morning).

If you have been running **Keystroke** at a single location, you can use those data files to help set up the Master data, as well as the data for the remote stores. If this is a new installation, the data files can be set up at one store and then copied to each subdirectory.

One of the most important assumptions in the design of MultiStore is that each store is very similar (preferably identical) in regards to the Inventory items available. In other words, any given Stock Number must represent the same item in each store. (See also Variations & Restrictions, later in this Appendix.)

Once the QOH information is set up for each store, it should *never* be manually adjusted – it should only be changed through transactions or the Variance function. You do not have to immediately run a polling procedure after setting up QOH information for each store, but you must poll all transactions from this point forward so that the Master store information will match that of the remote stores.

The Polling Script you will use must be customized for your business by your local Authorized **Keystroke** dealer to accommodate your desired MultiStore configuration. This includes information such as number of locations, telephone numbers, and data directories. Polling Scripts consist of a combination of standard DOS commands (e.g., **COPY**, **CD**, etc.), the MultiStore utility commands (Extract, StoreMerge, Update Inventory, and CopyNum), and special commands (e.g., Call/Dial, Transfer/Send, Hang up, etc.) which vary with the communication software/process you are using (e.g., CloseUp, pcANYWHERE, email, or FTP).

Sample Polling Scripts for dial-up, email and FTP are included towards the end of this Appendix; several variations of Polling Script files are also available through **Keystroke** Technical Support Services, on the **Keystroke** web site ([www.KeystrokePOS.com](http://www.KeystrokePOS.com)), and on the CD or disk accompanying the MultiStore program files.

### **Extract (EXTRACT.EXE)**

The Extract program is a stand-alone utility that extracts *new* transactions, customers, and/or serial numbers from your current data files into the file **KSTRN001.MRG**. This file can then be used with the StoreMerge program to merge this new information into another set of data files.

“New” transactions are those that have a date within the range of Extract’s Start and End date. If these dates aren’t specified (and they usually are not) the period will be from the last time Extract was run through the previous Midnight. Transaction data includes Sales Invoices, Orders, Layaways, Purchase Orders and Invoices, Credit Memos, Audit entries, Variances, and Time Clock entries.

“New” Customers are those that have a Last Purchase or Last Payment date after the starting date of the Extract period. To make sure all new Customers are brought over, make sure one of these fields is blank (i.e., does not contain a 0) in the Default Customer screen in the Database Manager.

The following switches may be used in addition to the standard **Keystroke** switches. Please review the document **EXTRACT.DOC** in the **DOC\** subdirectory for the most current information regarding switches that may be used with the Extract program.

**/M – Merge File Directory:** Use this switch if the file KSTRN001.MRG is to be made in a directory other than the program directory (e.g., /M C:\DATA).

**/NOLOCK – No Lock:** The program will not update the Transaction Lock date.

**/NODISPLAY – No Display:** The program will not display anything on the screen while it is running. This is used to speed up the program when running Extract remotely during automatic polling.

**/NOCUST – No Customer Update:** Use this switch if you do not want to extract Customer information. (This is usually used when merging into the Master store and you are not sharing Customer information amongst the stores; the same as using EXCLUDE=C.)

**/NOVAR – No Variance:** Use this switch to disable the extracting of Variance transactions. (This is the same as using EXCLUDE=0.)

**/NUMBERS – Numbers:** Extract serial numbers from Inventory. (This is the same as putting INCLUDE=s in the EXTRACT.DAT file.)

**/FORCEDATE – Force Ending Date:** Use this switch to save the End Date of the Extract as the new Start Date, even if it is before the last Extract's End Date.

**START={date}:** This switch is used to change the starting date of the Extract period (e.g., START=04/10/02 will cause the program to start at 0:00 a.m. on 04/10/02).

**END={date}:** This switch is used to change the ending date of the Extract period (e.g., END=04/11/02 will cause the program to end at 0:00 a.m. on 04/11/02). Note that the actual ending time is the previous Midnight. Setting END=M represents the nearest Midnight to the current time.

**STARTTIME={time}:** This switch is used to change the starting time of the Extract period (e.g., STARTTIME=5:30a will cause the program to start at 5:30 a.m.).

**ENDTIME={time}:** This switch is used to change the ending time of the Extract period (e.g., ENDTIME=18:15 will cause the program to end at 6:15p.m.).

**EXTRACT={CV1WQ05RYyA0T9s}:** This switch is used to define what category of data is to be extracted. Enter a string of one or more characters (C=Client, V=Sales Invoice, 1=Sales Order, W=Layaway, Q=Quote, O=Purchase Invoice, 5=Purchase Order, R=ReservedOrder, Y=SalesPayment, y=PurchasePayment, A=ROAPayment, 0=Variance, T=TimeClock, 9=Audit, s=SerialNumbers). The default is all ON: EXTRACT=CV1WQ05RYyA0T9 (s is not ON by default).

**INCLUDE={CV1WQ05RYyA0T9s}:** This switch is used to add additional characters to the list of categories to extract (e.g., INCLUDE=s to extract SerialNumbers).

**EXCLUDE={CV1WQ05RYyA0T9s}:** This switch is used to remove characters from the list of categories to extract (e.g., EXCLUDE=C to not extract Customers).

---

***NOTE:** These switches can also be specified in the file EXTRACT.DAT. In the last 3 switches above, upper/lower case DOES make a difference; therefore, you must use the EXTRACT.DAT file to specify SerialNumbers and PurchasePayments (s and y), since command line switches are always passed to the program as all uppercase characters.*

---

The following parameters are all optional and are used to establish a range (or filter) of which records to extract. If any switch is not used, the records will not be tested against it. If a **Start=** switch is used, the record's value must be greater than or equal to the value specified; if an **End=** switch is used, the record's value must be less than or equal to the value specified (e.g., **STARTSTORE=2 ENDSTORE=2 STARTREG=4** will only extract records whose Store Number is 2 and whose Register Number is 4 or higher).

**STARTSTORE={Starting Store Number}**: Specify the starting Store Number of the range of records you wish to extract.

**ENDSTORE={Ending Store Number}**: Specify the ending Store Number of the range of records you wish to extract.

**STARTREG={Starting Register Number}**: Specify the starting Register Number of the range of records you wish to extract.

**ENDREG={Ending Register Number}**: Specify the ending Register Number of the range of records you wish to extract.

**STARTCLERK={Starting Clerk Number}**: Specify the starting Clerk Number of the range of records you wish to extract.

**ENDCLERK={Ending Clerk Number}**: Specify the ending Clerk Number of the range of records you wish to extract.

**STARTTRANS#={Starting Transaction Number}**: Specify the starting Transaction Number of the range of records you wish to extract (not supported by all categories).

**ENDTRANS#={Ending Transaction Number}**: Specify the ending Transaction Number of the range of records you wish to extract (not supported by all categories).

Unless specified not to (with **/NOLOCK**), the Extract program will cause the Transaction Lock Date/Time to be updated to the End Date/Time of the Extract period (usually the current date at 0:00 a.m.). For more information on the Transaction Lock Date/Time, see Configuration Manager – Global Parameters in the main User Guide.

Running Extract modifies the file **EXTRACT.DAT** by updating the **START=** and **STARTTIME=** switches. Other switches can be included in the file and will be used if the same switch is *not* used as a command line switch (command line switches take precedence over any settings in **EXTRACT.DAT**).


## **StoreMerge (STOREMRG.EXE)**

The StoreMerge program is a stand-alone utility that merges transactions from the file **KSTRN001.MRG** (created by the Extract utility) into the current data files. As it merges the transactions, it will update Inventory and Customer information to correspond with the new transactions. It will also pull in new and changed customers.

---

***NOTE:** After each transaction is merged in, it is marked so that it cannot be merged again. Therefore, if you want to merge with this KSTRN001.MRG file again you should make a backup copy first. This is done in all standard polling scripts.*

---

To run the program, first make sure the file **KSTRN001.MRG** is in the **KEYSTROK\** directory. From a DOS prompt, type **STOREMRG**  (if you run **Keystroke** with any switches, include them in this command). The program will remind you that you should back up your data files before proceeding. It will then proceed to merge in all the new transactions and Customers. When it is through, the program will return you to a DOS prompt.

The following switches may be used in addition to the standard **Keystroke** switches. Please review the document **STOREMRG.DOC** in the **DOC\** subdirectory for the most current information regarding switches that may be used with the StoreMerge program.

**/D – Data Directory:** Use this switch to merge into a directory other than the Master (DATA\).

**/M – Merge File Directory:** Use this switch if the file KSTRN001.MRG is in a directory other than the program directory.

**/NOP – No Pause:** The program will run without user intervention. This should be used if StoreMerge is being run automatically by a script or batch process.

**/NOINV – No Inventory Update:** Use this switch if you do not want Inventory information updated. (This is usually used when merging into the Master store).

**/NONEXTNUMS : No Next Numbers** – Do not update Next Number on transactions. (This does not work on the Customer Next Number, which is always updated; use COPYNUM.EXE to reset it.)

**/SALESONLY – Sales Information Only:** Use this switch if you do not want Purchase Transactions to be updated. (This is the same as using EXCLUDE=OR5y.)

**/NOCUST – No Customer Update:** Use this switch if you do not want Customer information updated. (This is usually used when merging into the Master store and you are not sharing Customer information amongst the stores, and is the same as using EXCLUDE=C.)

**/NOVAR – No Variance:** Use this switch to disable the merging of Variance transactions. (This is the same as using EXCLUDE=0.)

**/NOTIMECLOCK – No Time Clock:** Use this switch to disable the merging of Time Clock entries. (This is the same as using EXCLUDE=T.)

**/NOAUDIT – No Audit:** Use this switch if you do not want Audit information updated. (This is the same as using EXCLUDE=9.)

**/NUMBERS – Serial Numbers:** Use this switch if you want to merge in Serial Numbers. (Erases all current serial numbers, then adds back in those from the merge file, and is the same as putting INCLUDE=s in the STOREMRG.DAT file.)

**/LOWMEM – Low Memory:** Use this switch if you do not have enough conventional memory (i.e., if you normally use this switch when running **Keystroke**).

**EXTRACT**={CV1WQ05RYA0T9s}: This switch is used to define what category of data is to be merged. Enter a string of one or more characters (C=Client, V=Sales Invoice, 1=Sales Order, W=Layaway, Q=Quote, O=Purchase Invoice, 5=Purchase Order, R=ReservedOrder, Y=SalesPayment, y=PurchasePayment, A=ROAPayment, 0=Variance, T=TimeClock, 9=Audit, s=SerialNumbers). The default is all ON: EXTRACT=CV1WQ05RYA0T9 (s is not on by default).

**INCLUDE**={CV1WQ05RYA0T9s}: This switch is used to add additional characters to the list of categories to merge in (e.g., INCLUDE=s to merge SerialNumbers).

**EXCLUDE**={CV1WQ05RYA0T9s}: This switch is used to remove characters from the list of categories to merge in (e.g., EXCLUDE=C to not merge Customers).

---

***NOTE:** These switches can also be specified in the file STOREMRG.DAT. In the above 3 switches, upper/lower case DOES make a difference; therefore, you must use the STOREMRG.DAT file to specify SerialNumbers and PurchasePayments (s and y), since command line switches are always passed to the program as all uppercase characters.*

---

The following parameters are all optional and are used to establish a range (or filter) of which records will be merged in. If any switch is not used, records will not be tested against it. If a **start**= switch is used, the record's value must be greater than or equal to the value specified; if an **end**= switch is used, the record's value must be less than or equal to the value specified (e.g., **STARTSTORE=2 ENDSTORE=2 STARTREG=4** will only merge in records whose Store Number is 2 and whose Register Number is 4 or higher).

**STARTSTORE**={Starting Store Number}: Specify the starting Store Number of the range of records you wish to merge.

**ENDSTORE**={Ending Store Number}: Specify the ending Store Number of the range of records you wish to merge.

**STARTREG**={Starting Register Number}: Specify the starting Register Number of the range of records you wish to merge.

**ENDREG**={Ending Register Number}: Specify the ending Register Number of the range of records you wish to merge.

**STARTCLERK**={Starting Clerk Number}: Specify the starting Clerk Number of the range of records you wish to merge.

**ENDCLERK**={Ending Clerk Number}: Specify the ending Clerk Number of the range of records you wish to merge.

**STARTTRANS#**={Starting Transaction Number}: Specify the starting Transaction Number of the range of records you wish to merge (not supported by all categories).

**ENDTRANS#**={Ending Transaction Number}: Specify the ending Transaction Number of the range of records you wish to merge (not supported by all categories).

The file **STOREMRG.DAT** can be used to run StoreMerge using any of the above parameters. This file can be created using any DOS Editor, and can be used instead of startup switches to run StoreMerge using specified settings. Command line switches take precedence over settings in **STOREMRG.DAT**.

# Managing Inventory Data

## Inventory Link (INVLINK.EXE)

Inventory Link is a separate program included with **Keystroke**. It provides the ability to easily look up the current availability of an Inventory item in other stores. This function is accomplished through the use of relatively small data files that are periodically generated and transferred amongst the stores. The small size of these files minimizes the time required to transfer via modem, and in turn makes it easy and practical to provide each store with current data more frequently.

The Inventory Link program must be run from the **Keystroke** program directory, and requires a few initial setup steps to configure properly. When the program is run, it creates files named **KSIL#.DAT** (where # represents the Store Number, and is determined by the setup of the Store Number field in the Configuration Manager – Company Menu – Name/Address function). These files are also stamped with that store's **Keystroke** serial number. To ensure a proper serial number stamp, the file **KEYSTROK.LIC** must be copied into each store's subdirectory on the Master computer. This file can be found on the **Keystroke** license disk (or Disk 1 of the **Keystroke** installation disks, if installing from floppies), or in the **Keystroke** program directory at the remote store.

When running Inventory Link, you can use the **/D** switch to specify which data directory to use. For example, typing **INVLINK /D STORE2** will create **KSIL2.DAT**, which contains the QOH values for store number 2.

The **KSIL#.DAT** files are stored in the **Keystroke** program directory of the Master store. The entire set of files (i.e., **KSIL\*.DAT**), needs to be copied to the data directory at each of the remote stores through the polling function. Once each store has received the .KSIL files for the other stores, the Qty Lookup feature may be used to look up the Quantity On Hand at each of the other stores. This information is available anywhere within the program where an Inventory item is displayed by highlighting the item in question and pressing **SHIFT F5**. A dialog box will display each Store Number followed by the date and time of the last Inventory Link update and the respective Quantity On Hand.

---

**NOTE:** The Store Number must be different for each store, (i.e., **KSIL1.DAT**, **KSIL2.DAT**, etc.). Each store must also be using a separate licensed copy of the **Keystroke** program with a unique Serial Number.

---

The file **INVLINK.DAT** may be used to control which stores show up on the list of stores. A store may be excluded from the list; it may also be marked to display on the list, but not to be included in the Total (if all stores are excluded from the Total, the Total will not be displayed).

The **INVLINK.DAT** file is a text file in INI format, and can be created using any text editor (such as DOS EDIT or Windows NotePad). Each line of the file should contain a Store Number, followed by an equal sign, followed by the Name to be displayed. To exclude a store from the list, set the Store Name to OFF. To exclude a store from the Total, type a dash (i.e., hyphen or minus sign) in front of the Store Number. Any stores not listed will appear on the list, but will display the Store Number instead of the Name.

For example, a file that looks like this:

```
1=Long Beach
2=Dillon
3=OFF
-4=Warehouse
```

will not display Store 3. Store 4 will be displayed, but will not be added to the Total.

## Managing Store Data

### StoreLink Module

The StoreLink module is designed for businesses operating multiple retail store locations with nearly identical Inventories. StoreLink provides functions that concentrate on the management of Inventory in multiple stores and/or warehouse locations. These functions include: the ability to look up stock levels of individual items at alternate locations, composite Inventory reports listing Inventory levels of individual locations and totals, globally update Inventory items including pricing and descriptions, and the ability to transfer Inventory between locations.

The main screen in the StoreLink module may be used to enter Store Transfers (moving Inventory from one location to another). Access the StoreLink module by choosing Custom – Store Link from the **Keystroke** Main Menu. You must have a Security Level of 5 (Access Custom Modules) to access the StoreLink module. When exiting the StoreLink module (Exit Menu), the program will take you to the Logo screen and prompt for a Clerk Number/Password. This ensures that the data files are checked and that the proper Security Levels are used when switching between different stores. The Query Clerks function is also disabled if the Current store is changed while in this module.

---

**NOTE:** StoreLink data files are always stored in the DATA\ directory below the current program directory. Therefore, if you are running on a network, **Keystroke** must be run from the file server on any machine that is going to use the MultiStore utilities (i.e., don't install a separate copy of **Keystroke** on a local hard disk if that machine will be used to run StoreLink or the Inventory Update function). Transfers are saved in the files DATA\KSSTR\*.DAT; Print Button setup is stored in DATA\FORMS.DAT.

---

### Inventory Transfers

The main screen of StoreLink is used to transfer Inventory items between stores. Inventory Transfers are very much like Invoices created in the Sales and Purchase Manager modules; therefore, the procedures for creating an Inventory Transfer are almost identical to entering a sales transaction. Items are always transferred *from* the Current store *to* the Transfer To store. When a transfer is completed, the appropriate items are subtracted from the Current store and added to the Transfer store.

To enter an Inventory Transfer, first verify that the Current store is the store that Inventory is being transferred *from*. (To specify the Current store, use the Current function on the Store menu.) Press **F8** to select the store that Inventory is being transferred *to*. Then, enter the Inventory items and the quantities you wish to transfer.

After entering all the items to be transferred, press **ENTER** on the last item line and the Print Box will appear. Select the **[Print]** button to print a Transfer form, and **[Save]** to record the transaction and update the appropriate Inventory data.

When the Transfer is saved, the program will subtract the Inventory quantities from the Current store and add them to the Transfer To store. If the program can't find the Stock Number of an item in the Transfer To store data file, the Inventory record will be automatically copied from the Current store to the Inventory database of the Transfer To store.

Inventory Transfers can be edited, deleted, recalled, and reprinted, the same as Invoices and Purchase Orders, by using the appropriate menu functions. When a Transfer transaction is edited, deleted, or recalled, the program will adjust all Inventory levels accordingly.

Because the Store Transfer function adjusts Inventory quantities in two locations in one single step and does not require any intervention at each location, the normally desirable "checks and balances" are not available. In other words, there is no data input required to verify that Inventory actually reaches the location to which it has been transferred. Also, because the Inventory Transfer function requires that all data files be installed on a single machine and can only be done at the Master Location, many businesses instead choose the following alternate method of transferring Inventory between stores.

This simple work-around method consists of entering two Purchase Invoices (one at each location involved in the transfer). First, add a Vendor for all stores for which inventory might be transferred to or from. At the location where Inventory is being transferred *from*, the Purchase Invoice will include all items being transferred with *negative* quantities. Then, at the store where the Inventory has been transferred *to*, a complementary Purchase Invoice is created with the same items, but with *positive* quantities. This two-step transaction is essentially the same as if Inventory were received directly from a Vendor, and will adjust the Inventory levels appropriately for the transfer.

---

**NOTE:** *Since StoreLink allows you to simultaneously use Inventory files from more than one store, it is imperative that the Stock Numbers of all transferable items are identical for all stores. Any items that are unique to an individual store should be assigned Stock Numbers that will not conflict with common items, (i.e., those in the Master Inventory). For items that are not common between stores, it is suggested to use Stock Numbers below a certain number in order to avoid such conflict. For instance, you might assign Stock Numbers greater than 1,000 to all the items that are common to all stores, and use only Stock Numbers less than 1,000 for items that are unique to individual stores. If you do this, make sure to not use the Stock Numbers in the unique range in the Master Inventory, or the Inventory Update function will replace the unique items in each store with its own. This method can be very complicated, and is not recommended.*

---

# StoreLink Menus

The StoreLink menu bar consists of the common menu options, along with the menus and functions listed below. Note that the Exit Menu functions somewhat differently in the StoreLink module than in other areas of the program.

## Transfer Menu

**Clear** – remove the current Transfer from the screen

**Edit/Print** – edit an existing Transfer

**Copy** – copy an existing Transfer to a new one

**Delete** – delete an existing Transfer

**Recall** – recall a deleted Transfer

**Number/Date** – change the current Number and/or Date of the current Transfer

**Print/Save** – print and save the current Transfer

**Parameters** – enter default settings for Store Transfers and Updating Inventory

## Store Menu

**Current** – select the current Store

**Transfer To** – select the Store to Transfer Inventory to

**Setup** – edit list of available Stores

**Update Inventory** – add new items/update existing items from one Store to all other Stores

## Reports Menu

**Inventory** – print a list of Inventory levels for each/all Store(s)

**Transfers** – print a list of Store Transfers

## Special Menu

**Import Line Items** – import Inventory items from a file

**Pack** – pack the Transfer files

## Exit Menu

**The Exit menu in StoreLink will return you to the Keystroke Logo screen instead of popping up the Main Menu.** This is done so the program can check the data files to be used (see Store Menu – Current function below).

## Transfer Menu

### Clear Function

The Clear function is used to abandon unsaved information entered on the current Transfer screen so that a new one may be entered. This function clears data from the screen only, and does not affect information which has been previously saved, such as when viewing a Transfer via the Edit/Print function.

While on the main Transfer screen, the Clear function can be selected from the Transfer menu. A confirmation message will appear; select the **[Ok]** button to proceed in clearing the screen, or the **[Cancel]** button to abort the Clear function.

## Edit/Print Function

The Edit/Print function is used to look up previous Transfers for the purpose of viewing, editing, and/or reprinting. The Edit/Print function will prompt you to select the method by which the list of Transfers will be sorted: Number, Date, From Store Number, or To Store Number.

Following the selection of a sort category, a list of Transfers will be displayed. In this list, use the arrow keys and/or the Quick Search feature to locate and highlight the desired Transfer, then press **ENTER** to display the transaction. Take note of the Date Range displayed in the top border of the list box; the current file includes only Transfers dated within this range. If the Transfer you are looking for is not within the range of dates, you may need to switch to a different transaction data file by pressing **CTRL** **PG↑** or **CTRL** **PG↓**. When finished viewing or editing a Transfer, either save it as you would with a new Transfer, or use the Clear function to abandon the Transfer without saving changes.

### Printing Transfers

Printing a Store Transfer is the same as printing an Invoice, Quote, or other form. First you must set up a Print Button for Store Transfers (Configuration Manager – Tables Menu – Print Buttons function). Choose Store Transfer (at the bottom of the list) and press **F3** or **ENTER** to edit. Give it a name, make sure there is an “X” in the Active field, then choose the form file you wish to print and the printer you wish to print to. (By default, **Keystroke** ships with two forms with the **.STR** extension. As with all forms, these can be customized for you by **SBS** or by your local Authorized **Keystroke** Dealer.) For more information on setting up Print Buttons, see the Print Buttons section of the Main **Keystroke** User Guide.

---

***NOTE:** Print Buttons must be set up using the KEYSTROK\DATA\ directory. If your Master data is located somewhere other than this directory, the Store Transfers Print Button will not be displayed in the list of Print Buttons. You will need to start **Keystroke** without using any startup switches to initialize empty data files in this directory before continuing. Because Store Lists and Store Transfers are also stored in the KEYSTROK\DATA\ directory, we strongly recommend that you always set up the Master data in this directory.*

---

## Copy Function

The Copy function can be used to copy a previously saved Transfer to a new one. This function will copy all information from the selected Transfer to the current Transfer entry screen. The program will assign the current system Date/Time, and the next available Transfer Number to the new Transfer. The new Transfer may then be modified, printed, and saved.

## Delete Function

The Delete function is used to remove the Transfer from the Transfer list boxes. The Recall function (see below) can be used to restore Deleted Transfers to their original status; however, once the file in which a Deleted Transfer has been Packed (see the Pack function), it can no longer be Recalled.

When you select the Delete function, you must first choose a sort category by which to list the Transfers. After selecting a sort category, select which Transfer to Delete by positioning the highlight bar in the list box and pressing **ENTER**.


## ***Recall Function***

The Recall function is used to restore a Deleted Transfer. The Transfer list will contain only Transfers that have been Deleted since the last time the Pack function was run.

After selecting the Recall function, select which Transfer to Restore by positioning the highlight bar in the list box and pressing .



## ***Number/Date Function***

The Number/Date function can be used to change the Transfer Number, Date, and/or Time of the current Transfer. When a new Transfer is created, **Keystroke** automatically assigns the next sequential Transfer Number and the current system Date/Time. The Number/Date function can be used to set a starting Transfer Number (e.g., to set your first Transaction Number at “1001” rather than “1”), or to override the automatically assigned Number for other reasons.

After changing the Transfer Number and selecting the **[Ok]** button on the dialog box, the program will display the prompt “**Change number counter for future transfers?**”. Your options to respond will be **[Yes]**, **[No]**, or **[Cancel]**. Selecting **[No]** or **[Cancel]** indicates that you wish to change the Transfer Number assigned to the current Transfer only. Pressing **[Yes]** (or ) will use the new number as a starting point for numbering all future transfers.

The Date and Time may be changed for the current Transfer only. To change the system Date/Time, use the Date/Time function in the Configuration Manager.

## ***Print/Save Function***

The Print/Save function brings up the Print/Save dialog box. This is the same as pressing  with the cursor on the last empty line on the Transfer entry screen, or pressing .

## ***Parameters Function***

The Transfer Parameters function includes settings which provide control over how Store Transfers are entered. Parameters allow you to change what the screens will look like, what data entry fields the program will prompt you to use, and a variety of other default settings.

The Parameters function is divided into two sections: Entry and Inventory Update. Settings in the Entry Parameters section include settings to control how Transfers are entered; Inventory Update parameters determine how Inventory information is updated when the Update Inventory function is used.

## Entry Parameters

The Entry Parameters are primarily used to control how the system will prompt for information throughout the entry of a Store Transfer.



*Transfer Entry Parameters*

### **<Item Field>**

The <Item Field> setting determines the default entry mode for line items. This should be set to the Inventory field you will most commonly use to enter items. Options for this setting include: Description, Product Code, Stock Number, Class, and Manufacturer.

### **Enter Store First**

The Enter Store First setting causes the program to automatically prompt for the Number of the Store to which Inventory items will be transferred. This should always be left on since you must specify the store that the items are being transferred to before you can save the Transfer.

### **Confirm Qty**

The Confirm Qty setting determines whether the cursor will automatically go to the Qty column during the entry of each item. If this setting is enabled, the Qty must be confirmed (or changed) for each line item. If turned OFF, the Quantity will always be one (1), unless you press **TAB** to move to the Qty column manually.

### **No Out of Stock Transfers**

The No Out of Stock Transfers setting determines whether items with a Quantity On Hand of zero (0) or less may be transferred between stores. If this setting is enabled, the program will only allow items to be transferred if the QOH of the item is greater than zero (0).


### **Confirm Cost**

The Confirm Cost setting determines whether the cursor will automatically go to the Cost column during the entry of each item. If this setting is enabled, the Cost must be confirmed (or changed) for each line item. You may also press **TAB** to move to the Cost column manually.

### **Copy Inventory Comment**

The Copy Inventory Comment setting causes Comments on Inventory database records to be automatically copied to the line item Comment field on the Transfer. In the Inventory database Comment box, text which is enclosed in vertical bars (or following a single vertical bar character in the Comment box) will *not* be printed on the transfer form (e.g., **text to be printed |text not to be printed|**).

### **Default to Save**

The Default to Save setting causes the program to automatically select the Save button when proceeding to the Print box. To Save the Transfer and start a new Transfer, press **[Save]**; otherwise, use the arrow keys to select the print button (the name of this button will depend on the name you gave it when you set it up in the Configuration Manager), then press .

---

***NOTE:** Ordinarily, the Print box displays two print buttons: Print and Save. The Print button is set up in the Configuration Manager – Tables – Print Buttons – Store Transfer. If only the Save button is displayed, it is probably because the Master Store data is not located in the DATA\ directory. For this reason (among others), we recommend that the Master Store data always be located in the DATA\ directory. To access this directory, start **Keystroke** without any startup switches.*

---

### **Print and Continue**

The Print and Continue setting causes the program to automatically prepare for the next Transfer following the selection of any button in the Print box. If the Print and Continue setting is disabled, the **[Save]** button must also be selected in order to prepare for the next transaction.

### **Update Last Cost**

The Update Last Cost setting determines whether the Last Cost field on Inventory database records in the receiving store will be updated. If this setting is enabled, the program will copy the Cost field from the Store Transfer to the Inventory record of the store to which the item is being transferred.

### **Update Avg. Cost**

The Update Average Cost setting determines whether the Average Cost field on Inventory database records will be updated when items are Transferred. If this setting is enabled, the program will calculate a new Average Cost for each item Transferred and copy that amount to the Average Cost field on the respective Inventory records. For a complete discussion on the formula used to calculate Average Cost, please refer to the Update Average Cost topic in the Purchase Manager section of the main User Guide.

### **Don't Update Master Store QOH**

This setting determines whether the QOH field on Inventory database records will be updated at the Master store when items are Transferred. If this setting is enabled and a Transfer is saved, the program will not add or subtract Quantities On Hand in the Master Store's Inventory (Store Number 0). This parameter should *only* be used if you want the Master Store's Inventory to be a consolidation of Inventory for all stores, or a Virtual Store, and should *not* be used if you are using the Master Store for purchasing and distribution of Inventory.

---

***NOTE:** The Master Store must be Store Number 0 for this parameter to work.*

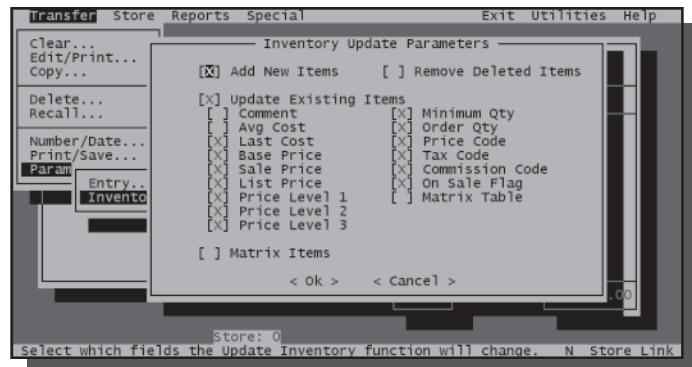
---

### **Reset Avg. Cost**

The Reset Average Cost setting determines whether the Average Cost field on Inventory database records will be reset when items are Transferred. If this setting is enabled and items are Transferred to a store with QOH less than zero, the program will copy the current Cost to the Average Cost. Otherwise, the new Average Cost will be appropriately calculated. For a complete discussion on the formula used to calculate Average Cost, please refer to the Update Average Cost topic in the Purchase Manager section of the main User Guide.

## **Inventory Update Parameters**

Inventory Update Parameters include settings that determine what fields are updated and what methods are used to control Costs, Prices, and other Inventory database information when the Update Inventory function is run.



*Inventory Update Parameters*

### **Add New Items**

The Add New Items setting determines whether Inventory items from the Master database will be added to the database of the receiving store if that item does not already exist in that store's database when Update Inventory is run. Turning ON this parameter will cause the program to add new items to all stores being updated.

### **Remove Deleted Items**

The Remove Deleted Items setting determines whether Inventory items deleted from the Master database will also be deleted from the databases of the other stores when Update Inventory is run. Turning ON this parameter will cause the program to look at the Recall list to determine which items have been deleted, and delete those items from the databases of all stores being updated.

---

**NOTE:** *Packing the database clears the Recall list, so make sure to run Update Inventory before packing the Master Inventory database.*

---

### **Update Existing Items**

The Update Existing Items setting determines whether fields on the Inventory database record will be updated when Update Inventory is run. If this setting is OFF, items that already exist will not be changed. Turning ON Update Existing Items will change Description, Code, Pricing, etc. on Inventory records, but will not change QOH or Total Sold Quantities.

---

*NOTE: If Add New Items, Remove Deleted Items, and Update Existing Items are OFF, running the Update Inventory function will not do anything.*

---

### **Inventory Fields**

Designate the fields to be updated (when updating an existing item) by placing an “X” in the checkbox in front of the fields following the Update Existing Items setting. If an item is not checked (turned OFF), the field will not be updated, and will keep the information currently on that record.

The following fields *may* be updated: Comment, Avg Cost, Last Cost, Base Price, Sale Price, List Price, Price Level 1, 2, 3, Minimum Qty, Order Qty, Price Code, Tax Code, Commission Code, On Sale Flag, and Matrix Table.

The following fields are *never* updated: QOH, Qty Allocated, Qty On Layaway, Qty On Order, Total Qty Sold, Total Qty Purchased, Last Sold Date, Last Purchased Date, Last Printed Price, and Serial Numbers.

All other fields are always updated.

### **Matrix Items**

The Matrix Items setting determines whether the program will update Matrix Items as well as regular Inventory items. Turning ON this parameter causes the program to update **MTX\* .DAT** and **KSMT\* .DAT** (all Matrix data: Matrix Tables/Categories/Elements) from the Master Inventory database to the databases of all stores being updated.

## **Store Menu**

### **Current Function**

The Current Function is used to switch between “sets” of data files. The set of data files that **Keystroke** will use is determined by the “Current” store. When entering a transfer, the Current store is also the “Transfer From” store. When you first enter StoreLink, the Current store will be the store at the top of your Store Setup list (usually number zero (0), or “**Master**”). To use a different set of data files, select Current from the Store menu, then select which store’s data files you want to make Current. After selecting a store, its data files will become the Current files for the program. This means if you leave StoreLink, all the information will be based on the Current store’s data files. Thus, you can print reports, review invoices, or just about anything else as if you entered **Keystroke** using that store’s data set.

The Current function is usually only used for Store Transfers. We recommend you set up separate icons and/or menu items to enter **Keystroke** using each store’s data set. We also recommend you set up different color schemes to help you keep track of which data set you are currently using.

---

**NOTE:** When using StoreLink, be aware that the Clerks are probably different for each store. If you select the Change Clerk function, the only available choices will be from the data files for the Current store. Thus, if you do not know a valid password, you could be locked out of the system and not be able to continue. To avoid this situation, add a common “System Supervisor” with a Security Level of zero (0) to each store’s Clerk database.

---

## Transfer To Function

The Transfer To function is used to specify to which store Inventory items will be transferred. Highlight the store you want items to be Transferred To and press **ENTER**. You can also enter the Transfer To store on the main transfer screen. Use **F8** to switch between line item entry and the Transfer To field.

## Setup Function

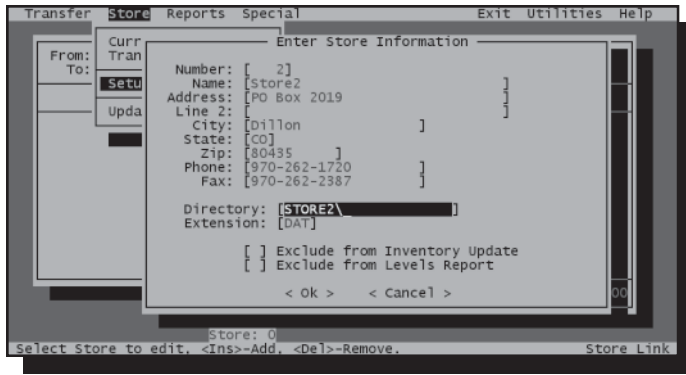
The Setup function is used to set up information about each of your stores. This information includes much of the same information as that entered in the Configuration Manager – Company Menu – Name/Address Function. This information is used by the program for the Update Inventory function, StoreLink reports, and Inventory lookup at the Master location. The list of stores is stored in the data file **KSSTORES.DAT**, which is always located in the **DATA\** subdirectory.

### Setting Up StoreLink

The first step in setting up StoreLink is to create a set of data files for each store. The recommended way of setting up your stores is to use the “Master” store as your “warehouse” or “consolidation” store. This is where you will create new and edit current Inventory items, as well as to manage your Accounts Receivable. Each store’s data files should be kept in separate directories below the **Keystroke** program directory (e.g., **STORE1\**, **STORE2\**, etc.).

When you first enter the StoreLink module, the program will create a store called “Master”. This store will have the extension **.DAT** and will be stored in the **DATA\** directory below the **Keystroke** program directory.

After entering the StoreLink module, select the Setup function from the Store menu. The program will display a list of all the stores currently entered in the system (this will be just the one store, “Master”, the first time you enter the Setup function). This list is used to add, edit, or delete stores. Press **INS** to Add a store. To Edit a store, highlight the store and press **ENTER** or **F3**; to delete a store, highlight the store and press **DEL**. To place the Master Store at the top of the list, highlight it and press **SHIFT DEL**, then move the cursor to the top of the list and press **SHIFT INS**.



When entering information for a new store, make sure the directory and extension for data files are correct.

StoreLink – Setup Function.

Each store will automatically be assigned a Store Number. You should change the number here and assign the appropriate Store Number in the Configuration Manager – Company function for each store. Be sure the numbers in the Store Setup list match the number stored in the Configuration Manager (Company Name/Address function) of each data set. If these are not set up properly, the Inventory Link feature will be confusing, and the Update Inventory function could overwrite valid data with invalid data.

---

**NOTE:** If you do not specify a separate directory for each store's data, by default the data will be located in the **KEYSTROK**\ program directory. (This is **NOT** recommended.) It is important to make sure each store has a separate directory from the other stores.

---

Most of the fields on the Enter Store Information setup screen are self-explanatory; however the following fields merit additional explanation:

- Exclude from Inventory Update** – If this setting is marked ON, this store will not be included in the Update Inventory function (i.e., will not update Inventory records with changes which have been made in the Master store Inventory).
- Exclude from Levels Report** – If this setting is marked ON, QOH information for this store's Inventory items will not be included in Inventory listings generated by the StoreLink – Reports Menu – Inventory function.

---

**NOTE:** The first store on the list is considered the “Master” for the Update Inventory function. If this store is marked to be Excluded from this function, the program will use the first store on the list that is not marked as Excluded for the Master Inventory.

---

## Update Inventory Function

The Update Inventory function may be used to update Inventory records with changes which have been made in the Master store Inventory. This function may be used to add new items, update existing items (this will correct Description, Code, Pricing, etc. but will not change QOH or Total Sold), or removed deleted items. How the Update Inventory function works is determined by the Inventory Update Parameters settings (Transfer Menu – Parameters Function), described previously.

---

**NOTE:** The first store on the list (usually the “Master” – Store 0) controls the Update Inventory function.

---

# Reports Menu

## Inventory Function

StoreLink provides Inventory Listing reports which list the Quantity On Hand levels of each store for all Stock Numbers in the Inventory file of the Master store. The options on the report setup screen are similar to those used in the Report Manager module.

**<Sort By>** – *Description, Product Code, Stock Number, Class, or Manufacturer (with Range)*

**<Group By>** – *None, Department, or Category*

**In Stock Only** – *include only those items with total (for all Stores) Quantity On Hand not equal to zero*

**Include Service Items** – *include those items marked as Service (if items are not appearing on the report that should, try checking this option)*

**Add Allocated to QOH** – *add number of Allocated items to total Quantity on Hand*

**Add Layaway to QOH** – *add number of Layaway items to total Quantity on Hand*

**Print Class** – *show Class for each item (instead of Product Code)*

**Use InvLink Files** – *If turned ON, the report will use Store QOH data from any KSIL\*.DAT files (created by the INVLINK.EXE utility) that exist in the data directory. If the program has access to both the complete Inventory files and the KSIL file for a particular store, the data in the KSIL file will take precedence*

**<Filter>** – *Filter by data fields available on Inventory records*

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

## Transfers Function

Provides information on Transfer activity.

**<Form>** – *Select the style of the Report you want to print (list is determined by files beginning with MT that have the extension .KSR in the REPORTS subdirectory)*

**<Filter>** – *Filter by data fields available on Transfers*

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

# Special Menu

## Import Line Items Function

The Import Line Items function is similar to the same function in the Sales Manager, and is used to import line items from an ASCII quote/comma delimited file. Such a file might typically be obtained from using a portable barcode data collector, custom transaction entry device, or perhaps from another custom software application. Data read using this function is added to the Store Transfer screen, the same as if the items were entered manually. Running the Import Line Items function twice on the same transaction will add additional line items to the bottom of the transaction, or will duplicate quantities if the **<Merge Same Items>** setting is used.

Selecting the Import Line Items function will pop up a dialog box which will prompt you to enter the name of the file containing the data to import. The default file name is **COUNT.TXT**; you can change this to reflect the name of your source file.

The **<Merge Same Items>** setting determines how the program will handle duplicate items. This parameter allows the program to check each item as it is entered against items already on the Transfer. If a duplicate item is entered, the program can merge the items together (add the quantities entered on the new item to the quantities of the one that already exists). The parameter can be set as follows:

**No:** Don't check for duplicate items on the Transfer.

**All Items:** Merge all duplicate items on the Transfer together.

**New Items Only:** Only check for duplicate items within the new items being Imported, and merge those items together.

## Pack Function

The Pack function is used to rebuild Store Transfer data files (**DATA\KSSTR\*.DAT**) and permanently remove deleted records. The Pack function first renames the file(s) to be Packed to the **.TMP** extension. It then reads data from the **.TMP** files and rewrites the data into its original extension (**.DAT**). As this procedure is performed, any non-data that is encountered in the file is discarded. The Pack function will recover any recognizable data from damaged files.

If something goes wrong while the Pack function is running (e.g., power or hardware failure), the **.TMP** files will remain in their original state. Pressing **[ESC]** before the Pack function is complete will cause the program to prompt whether you wish to cancel the Pack; pressing **[Ok]** will automatically copy the **.TMP** files back to their original file extension.

When you select the Pack function, the program will display a message reminding you to make a backup copy of your data files before proceeding. If you choose to ignore this warning message, you are risking the possibility of losing all your Store Transfer data. Following the warning message, the program will prompt you for the Approximate Date of the file you wish to pack. Enter a date and press **[ENTER]** to Pack.

# Miscellaneous/Error Checking Utilities

## **ChangeNums (CHGNUMS.EXE)**

The ChangeNums program can be used to renumber the Customer and/or Inventory databases. This may be required 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 can be used along with the Export function and the Importer Module 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. 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. Then run ChangeNums to make the actual Numbers agree with this new physical order.

The ChangeNums program may be used with the /D switch to change record numbers in other data sets.

**Example:**            CD\KEYSTROK  
                         CHGNUMS.EXE /DSTORE1

---

**NOTE:**     *DO NOT run ChangeNums if you are using the Matrix Manager, Production or Script Editor modules! Because order is important for these items, renumbering your Inventory items will cause you to lose the link between them and the main Inventory Item.*

---

## **CheckNum (CHKNUM.EXE)**

The CheckNum program is a stand-alone utility that checks to make sure that the two specified files have the same Next Number stored in them. The files must be **Keystroke** database files (Inventory, Customer, Vendor, or Clerk) to work properly. The syntax is:

**CHKNUM FILE1 FILE2 [UPDATE]**

If the Next Number in each file is different, the program will return one of the following DOS ErrorLevels:

- 0** – OK (Next Number is the same in both files)
- 1** – Number if File1 is HIGHER than in File2
- 2** – Number in File1 is LOWER than in File2
- 3** – Bad/Missing File Names
- 4** – Bad Files (size is less than 2048, or is not a multiple of 128)

The results are written to the file **CHKNUM.LOG**.

For example, the following batch file tests Customer files before copying them:

**Example:**            CHKNUM.EXE DATA\KSCST.DAT DATA\HOLD\KSCST.DAT  
                         IF ERRORLEVEL 1 GOTO BAD  
                         :GOOD  
                         COPY DATA\HOLD\KSCST\*.DAT DATA\KSCST\*.DAT  
                         GOTO EXIT  
                         :BAD  
                         ECHO Bad set of files, not copied  
                         :EXIT

If the optional Update parameter is used, the program will copy the numbers from FILE1 to FILE2 and will not generate Error Level 1 or 2. It will still print the comparison of the original two numbers in the log file.

**Example:** CHKNUM DATA\KSCST.DAT DATA\HOLD\KSCST.DAT UPDATE

### ***CheckScript (CHKSCRPT.EXE)***

The CheckScript program is a stand-alone utility that checks to see if each Inventory item has a Script attached to it. If an item possesses a Script, the program updates the hidden flag on that item (i.e., verifies Scripts are available). This program should be run if Scripts are copied from one data set to another. Update Inventory does update this hidden flag if the Scripts were created in the Master database.

**Example:** CD\KEYSTROK  
COPY DATA\KSSCR\*.DAT STORE1\KSSCR\*.DAT  
COPY DATA\KSSCR\*.DAT STORE2\KSSCR\*.DAT  
CHKSCRPT.EXE /DSTORE1  
CHKSCRPT.EXE /DSTORE2

### ***CopyClock (COPYCLK.EXE)***

The CopyClock program is a stand-alone utility that copies the TimeClock (Last In/Last Out) information from the file **KSCLK.TMP** to the file **KSCLK.DAT**. The times will only be copied if they are newer than the present times. CopyClock is used when the Clerk database is maintained at one location (usually the Master store). The data is then copied to the satellite locations. This program supports the standard **Keystroke** command line switches; it also supports the following special switch:

**/M – Merge File Directory:** Follow this switch with the location of the KSCLK.TMP file.

**Example:** CD\KEYSTROK  
COPY STORE1\KSCLK.DAT STORE1\KSCLK.TMP  
COPY DATA\KSCLK\*.DAT STORE1\KSCLK\*.DAT  
COPYCLK.EXE /D STORE1 /M STORE1

### ***CopyNum (COPYNUM.EXE)***

The CopyNum program is a stand-alone utility that copies the next available Customer number from **KSTRN001.MRG** to **KSCST.DAT** (The Customer database file). (Extract places the next available number in the merge file when it is extracting new Customers). The program first verifies that the two files exist, then copies the information. It is recommended that you run CopyNum after copying Customer files to each subdirectory.

It is recommended that each individual store have different starting numbers for all database and transactions (e.g., Customer Number or Invoice Number). These numbers are set via the Next Number function from the Database Manager's Setup menu or the Number/Date function in the Sales Manager. Since most stores share a common Customer database, the CopyNum program is used to restore the individual store's next Customer number after the master Customer database has been copied to it and before being copied out to the remote store(s).

This program supports the standard **Keystroke** command line switches; it also supports the following special switch:

**/M – Merge Directory:** Follow this switch with the location of the KSTRN001.MRG file.

```
Example:      CD\KEYSTROK
              COPY DATA\KSCST*.DAT STORE1\KSCST*.DAT
              COPYNUM.EXE /DSTORE1 /MSTORE1
              COPY DATA\KSCST*.DAT STORE2\KSCST*.DAT
              COPYNUM.EXE /DSTORE2 /MSTORE2
```

### **DateName (DATENAME.EXE)**

The DateName program is a stand-alone utility that changes the name of a file to a name representing the current date (in YYMMDD format). If a file already exists with that name (i.e., the same date), the program will append a dash and a letter (e.g., **-A**, **-B**, etc.) to the end of the file name.

The program will create a file named **DATENAME.TXT** that contains the names of the files created. This utility can be used to provide a backup for each day's **KSTRN001.MRG** file.

```
Example:      CD\KEYSTROK\STORE1
              ..\DATENAME.EXE KSTRN001.MRG *.MRG
```

### **FindFile (FINDFILE.EXE)**

The FindFile program is a stand-alone utility that finds any files containing the designated text. The syntax is:

```
FINDFILE FileMask SearchText
```

FileMask is the file name, which can include wildcards (e.g., **\*.DAT**); SearchText is the text you are searching for, and is not case sensitive. The program will display the names of the files containing a match to the SearchText, and will return one of the following DOS ErrorLevels:

- 0** – No Match
- 1** – Bad Syntax
- 2** – 1 or more files with a match

The output of the DOS ErrorLevel can be redirected to a file instead of the screen by using the parameter **>FileName** at the end of the command line.

```
Example:      CD\KEYSTROK
              DEL CHKSTAMP.LOG
              CHKSTAMP.EXE DATA\HOLD\KSINV*.DAT
              FINDFILE.EXE CHKSTAMP.LOG BAD
              IF ERRORLEVEL=1 GOTO BADINV
              COPY DATA\HOLD\KSINV*.DAT DATA\KSINV*.DAT
              :BADINV
```

## **GetMail (GETMAIL.EXE)**

Get Mail is an executable **Keystroke** program which will read entries from the Inbox of the specified MAPI server. Entries that match the specified parameters will have their attachments saved to the specified path and will then be deleted.

This program is designed to be used in e-Commerce applications, as well as MultiStore polling procedures. GetMail can be run from any directory.

The settings for Get Mail are controlled through the file **GETMAIL.INI**. The only command line switch supported is **INIFILE=** to specify a different file than **GETMAIL.INI**.

The following .INI file parameters are supported:

**NoPause=0/1**. If turned ON (e.g., NoPause=1), the program will not stop to display any messages (batch mode).

**UserName=Name**. Specify the name (or profile) of the MAPI Server to use (e.g., UserName = Microsoft Outlook).

**Password=Password**. – Specify if needed for selected MAPI Server.

**Subject=Filter**. Specify a 'Starts With' filter. The first characters of the email's subject must match in order for the program to save the attachment (leave blank to match all). If the AddSubjectToPath parameter is turned ON, the characters in the email's subject following the match will be used as subdirectory name when saving attachments (to save files to STORE1\, STORE2\, etc.). The subdirectory must already exist.

**NoteText=Filter**. Specify a 'Starts With' filter. The first characters of the email's message must match (leave blank to match all). Note: some email programs do not return correct text; in which case this must always be left blank.

**AddSubjectToPath=0/1**. If turned ON (e.g., AddSubjectToPath=1), the characters on the right of the Subject information will be included in the path to save the attachments.

**AttachmentPath=Name**. Subdirectory to save attachments to (will be below Program Directory, plus right-hand characters of email's Subject line if AddSubjectToPath is ON) (e.g. AttachmentPath=HOLD\).

**SaveTo=FileName**. Full FileName (including path) of where to save attachments. If this parameter is used, the AddSubjectToPath and AttachmentPath parameters will be ignored (e.g., SaveTo=C:\KEYSTROK\ORDERS\NEWORDER.TXT).

---

*NOTE: If the file already exists that the attachment should be saved to, the Email message will be skipped. This leaves it in your In Box, where it can be accessed again after the existing file is removed.*

---

**Example:** START /W GETMAIL.EXE INIFILE=GETINV1.INI /NOPAUSE

## ***Keystroke Polling (KSPOLL.BAT)***

The **Keystroke** Polling utility is a batch file that uses KSSCHDLR.EXE to start a polling process.

**Example:**           KSPOLL.BAT 10:30PM

The above line will run KSScheduler to pause the batch file until 10:30pm, at which time the batch file will continue.

```
Example:           ECHO OFF
                      KSSCHDLR %1
                      IF ERRORLEVEL 1 GOTO END
                      DIR KSPOLL.BAT >START.TXT
                      Echo Place call to Polling Routines Here
                      DIR KSPOLL.BAT >END.TXT
                      PAUSE
                      :END
```

## ***KSScheduler (KSSCHDLR.EXE)***

The KSScheduler utility is a simple DOS scheduler program which can be used in conjunction with DOS batch programs to cause your polling process or other programs to automatically run at a preset time of day. See **KSPOLL.BAT** above for an example of how to use KSScheduler. For the Windows version of **Keystroke**, we recommend using the Windows Scheduler or other available scheduling program.

---

**NOTE:** Use *START /W* within a batch file to run a 32-bit application and wait until it has closed.

---

```
Example:           KSSCHDLR.EXE 10:00PM
                      IF ERRORLEVEL 1 GOTO END
                      Echo Place call to Polling Routines Here
                      :END
```

## ***RemDupes (REMDUPES.EXE)***

The Remove Duplicates (RemDupes) program is a stand-alone utility that deletes any duplicate Sales Invoices or ROA payments, and is mainly used whenever the same set of sales data is accidentally merged in more than once (either through the StoreMerge or RecMerge programs). The transactions must be exact duplicates. If the same transaction is simply entered twice, they would not be duplicates as far as the program is concerned, because the Time Stamps would never be the same. The Remove Duplicates program works the same as if you deleted each entry manually.

```
Example: (Store1)    CD\KEYSTROK
                          REMDUPES.EXE /DSTORE1
```

Or

```
Example: (Master)   CD\KEYSTROK
                          REMDUPES.EXE /NOINV
```

## **SendMail (SENDMAIL.EXE)**

Send Mail is an executable **Keystroke** program which will post a message (with a file attachment) to the Outbox of the specified MAPI server. This program is designed to be used in e-Commerce applications as well as MultiStore polling procedures.

The settings are controlled through the file **SENDMAIL.INI**. The only command line switch supported is **INIFILE=** to use a different file than **SENDMAIL.INI**. The following .INI file parameters are supported:

**UserName=** *Name*. The name (or profile) of the MAPI Server to use (e.g., UserName = Microsoft Outlook).

**Password=** *Password*. If needed for selected UserName.

**SendTo=** *Text*. Address to send email to.

**Subject=** *Text*. Text to place on the subject line of the email.

**NoteText=** *Text*. Text to place in the body section of the email.

**Attachment=** *File*. Name (including path) of file to attach to email.

**Example:**            `START /W SENDMAIL.EXE INIFILE=SENDINV1.INI`

## **Stamp (STAMP.EXE) / CheckStamp (CHKSTAMP.EXE)**

The Stamp program is a stand-alone utility that will stamp the specified files with the current Date/Time by writing a serialized representation of the Date/Time when the program was run to the first record of each file. This stamp can then be used by the CheckStamp program to make sure all the files are from the same data set. The files must be **Keystroke** database files (Inventory, Customer, Vendor, Clerk, or Transaction) to work properly. This program is usually used to verify that a group of files (e.g., Inventory = **KSINV\*.DAT**) came from the same set of data files.

To use the Stamp/CheckStamp programs, specify a group of files (e.g., **STAMP.EXE DATA\KSINV\*.DAT**, or **CHKSTAMP.EXE DATA\KSINV\*.DAT**). The size of all files must be at least 2048 bytes and be divisible by 128 (all **Keystroke** Database and Transaction files meet these parameters) in order to be a valid file.

The CheckStamp program will write the results to the file **CHKSTAMP.LOG**. It will also return ErrorLevel 1 if any files don't match or ErrorLevel 2 if any files are not valid.

**STAMP Example:**            `CD\KEYSTROK`  
                              `STAMP.EXE STORE1\KSINV*.DAT`  
                              `STAMP.EXE STORE1\KSCST*.DAT`  
                              `STAMP.EXE STORE2\KSINV*.DAT`  
                              `STAMP.EXE STORE2\KSCST*.DAT`

```
CHKSTAMP Example:  CD\KEYSTROK
                   CHKSTAMP.EXE DATA\HOLD\KSINV*.DAT
                   IF ERRORLEVEL 1 GOTO BADINV
                   COPY DATA\HOLD\KSINV*.DAT DATA\KSINV*.DAT
                   :BADINV
                   CHKSTAMP.EXE DATA\HOLD\KSCST*.DAT
                   IF ERRORLEVEL 1 GOTO BADCUST
                   COPY DATA\HOLD\KSCST*.DAT DATA\KSCST*.DAT
                   :BADCUST
```

### ***StoreUpdate (STRUPDTE.EXE)***

The StoreUpdate program runs the Update Inventory function from the StoreLink module, and is usually used in polling scripts. The Store Setup and Update Inventory Parameters must be set prior to using this program. These parameters are stored in **KSSTORES.DAT** and **KSCNF.DAT** in the **DATA\** directory. See the StoreLink section and the Update Inventory function, earlier in this appendix.

```
Example:          CD\KEYSTROK
                  STRUPDTE.EXE
```

# Setting up MultiStore

---

**NOTE:** This setup assumes you are using the Main Store Master configuration, and that **Store#1** has been using **Keystroke** and the data is in the C:\KEYSTROK\DATA directory, and that **Store#2** is a new store. If you add more stores, simply repeat the steps from **Store#2** for **Store #3**, etc.

---

Get everyone out of **Keystroke** and stay out until you are finished with the setup procedures. Back up all critical data!

## Set Up Data Directories

- Create the following directories at the **Master** location:

```
C:\KEYSTROK\STORE1\  
C:\KEYSTROK\STORE2\  
C:\KEYSTROK\STORE1\HOLD\  
C:\KEYSTROK\STORE2\HOLD\
```

- Verify the Last Purchase and/or Last Payment date is blank on all the Customer Default screens (Customer Database – Setup Menu – Default Function). From the Database Manager, choose the Database Menu – Customer Database – Edit function. Then select Setup – Default and delete the “0” (zero) in either the LastPurchase Date or LastPayment Date. This causes the current date to be entered if a Customer record is added but neither a Purchase nor Payment is made.
- Enter an asterisk (\*) in the Tax Table field. This causes the Company Default Tax Table to be used for a Customer if they visit multiple stores.
- Copy \*.DAT from the DATA\ directory to the STORE1\ and STORE2\ directories, making a note of the date/time when the copy was made.

```
C:\KEYSTROK>copy data\*.dat store1\*.dat  
C:\KEYSTROK>copy data\*.dat store2\*.dat
```
- Edit/Save the file C:\KEYSTROK\STORE1\EXTRACT.DAT so the next Extract for **Store#1** will start from the date/time when files were copied (noted above after copy was completed). If the file does not exist, create one using the below format:

```
C:\KEYSTROK\STORE1>edit extract.dat
```

The **EXTRACT.DAT** File should be in the following format:

```
START=04/15/02  
STARTTIME= 0:00 am
```

- Edit the method of starting **Keystroke** to use the /D switch so Store#1 will be using data stored in the **C:\KEYSTROK\STORE1** directory. For example, the Shortcut Command Line for an icon in Windows 95/98 would read:

```
c:\keystrok\keystrok.exe /dstore1
```

The same syntax should be used for DOS Batch files.

### **Finish Setting Up Store 1**

While in **Store1**'s data (**KEYSTROK.EXE /DSTORE1**), perform the following important setup procedures:

- If not already done, set up the Company Name/Address/Store # information (Configuration Manager – Company Menu – Name/Address function). Update the Company Information and put a “1” in the Store Number field.
- Add a “System Administrator” Clerk with a Security Level of 0. A password is recommended, even if other Clerks are not using passwords.
- Set the Customer Next Number. While still on a Customer record in the Database Manager, select Setup – Next Number and change the next Customer number to a number not being used in **Store#2** (e.g., 10001). Remember, the highest Customer number that **Keystroke** can handle is 65,500.
- Optional: Set a similar Vendor Next Number in the Vendor database.
- Optional: Set the Transaction/Purchase Next Number in the Sales/Purchase Manager. (Although **Keystroke** does not require unique Invoice numbers, this is recommended to make it easier to recognize what store a transaction is from when viewing reports and other functions.) Use the Sales Manager – Transaction – Number/Date function to change the transaction number to a number not being used by **Store#2** (e.g., 100,000). When asked to “**Change number counter for future sales?**” answer “**Y**” (Yes). Do the same procedure in the Purchase Manager. **Keystroke** can handle transaction numbers up to 2 Billion (2,000,000,000), so don't be concerned about setting the Number to a relatively high value.

Once this step is complete, transactions can continue within the **Store#1** database while the remainder of the setup is completed.

### **Set Up Transaction Files**

- Delete and Initialize the transaction files (and supporting files) in the **Store#2** directory. The first step in this process is to delete the transaction history files from the **STORE2\** directory:

```
C:\KEYSTROK\STORE2>del kstrn*.dat (Transaction Files)
```

```
C:\KEYSTROK\STORE2>del kscrm*.dat (Credit Memo Files)
```

```
C:\KEYSTROK\STORE2>del ksbko*.dat (Back/Open Order Files)
```

- Delete the Inventory Files in the **Master** Data set (**DATA\**) and the **Store#2** (**STORE2\**) Data set. This is done so Quantity On Hand and Serial Numbers are reset.

```
C:\KEYSTROK\DATA>del ksinv*.dat (Master Inventory Files)
```

```
C:\KEYSTROK\STORE2>del ksinv*.dat (Store#2 Inventory Files)
```

- Once these procedures are complete, you are ready to initialize the data files. To initialize the **Master** Store's Inventory data files, type:

```
C:\KEYSTROK> keystrok.exe
```

Answer [Ok] when the “**DATA\KSINV.DAT not found! New file will be created**” message appears. Then answer [No] when asked to “**Initialize Startup Records...**”. When you are prompted to “**Enter Clerk Number:**”, press **[ESC]** to close the program.

- To initialize **Store #2**'s Transaction and Inventory data files, type:

```
C:\KEYSTROK> keystrok.exe /dstore2
```

Answer [Ok] when the “**STORE2\KSTRN.DAT not found! New file will be created**” message appears. Then answer [Ok] when the “**STORE2\KSINV.DAT not found! New file will be created**” message appears. Answer [No] when asked to “**Initialize Startup Records...**”. When you are prompted to “**Enter Clerk Number:**”, press **[ESC]** to close the program.

### ***Set Up StoreLink and use Update Inventory to Recreate Inventory Records***

- Start **Keystroke** and go into the StoreLink Custom Module (Custom – StoreLink). Go to the Store Menu, choose Setup, and add **Store#1** by hitting **[INS]** (or **[F2]**). Enter the name of the store, its directory (**STORE1\**) and data extension (**DAT**). Repeat this step to add **Store#2**.
- Edit the list of stores so that **Store#1** is on the top of the list. This will make **Store #1**'s Inventory be the **Master** for the Inventory Update function. Do this by using a Cut and Paste routine on **Store#1**:
  - Highlight **Store#1**.
  - Press **[SHIFT][DEL]** to Cut.
  - Move the cursor to the top of the list.
  - Press **[SHIFT][INS]** to Paste.
  - Press **[F10]** to Save the list of stores.
- Verify the Inventory Update Parameters (Transfer Menu – Parameters – Inventory Update Function). Both “Add New Items” and “Update Existing Items” should always be turned ON. It is recommended to check all boxes on this screen. Any items that are not checked will be assigned the value specified on the Default Inventory screen (Database Manager – Setup Menu – Default Function).
- Run Update Inventory from the Store Menu – Update Inventory Function. (Make sure you have another backup, since much time has already been spent setting up to this point!)

### ***Finish Setting Up Store 2***

While in **Store2**'s data (**keystrok.exe /Dstore2**), perform the following important setup procedures:

- Set up the Company Name/Address/Store # information (Configuration Manager – Company Menu – Name/Address function). Update the Company Information and put a “2” in the Store Number field.
- Add a “System Administrator” Clerk with a Security Level of 0. A password is recommended, even if other Clerks are not using passwords. (This Clerk should have the same Number/Name in all the stores.)
- Set the Customer Next Number. While still on a Customer record in the Database Manager, select Setup – Next Number and change the next Customer number to a number not being used in **Store#1** (e.g., 20001). Remember, the highest Customer number that **Keystroke** can handle is 65,500.
- Optional: Set a similar Vendor Next Number in the Vendor database.
- Optional: Set the Transaction/Purchase Next Number in the Sales/Purchase Manager. (Although **Keystroke** does not require unique Invoice numbers, this is recommended to make it easier to recognize what store a transaction is from when viewing reports and other functions.) Use the Sales Manager – Transaction – Number/Date function to change the transaction number to a number not being used by **Store#1** (e.g., 200,000). When asked to “**Change number counter for future sales?**” answer “Y” (Yes). Do the same procedure in the Purchase Manager. **Keystroke** can handle transaction numbers up to 2 Billion (2,000,000,000), so don't be concerned about setting the Number to a relatively high value.
- Back up the **Store2** data from the Master (**C:\KEYSTROK\STORE2\**) and restore it into the **DATA\** directory at the remote location.
- Create a **HOLD\** subdirectory underneath the **DATA\** directory at the remote location.

# Sample Polling Procedures

Rather than writing polling scripts from scratch, it is strongly recommended to begin with sample polling procedures. Sample polling procedures are shipped with the MultiStore module software (located in the **Keystroke** program directory). Other sample polling procedures are available from **Keystroke** Technical Support Services, and can also be downloaded from the **Keystroke** web site at [www.KeystrokePOS.com](http://www.KeystrokePOS.com).

The file MERGECU4.ZIP contains files necessary to perform polling using CloseUp v6.5 for DOS and Windows. The file MERGEAW4.ZIP contains files necessary to perform polling using pcANYWHERE for DOS; use MRGPCAW8.ZIP for pcANYWHERE version 8.0.2, and MRGPCA92.ZIP for pcANYWHERE v9.2 for Windows 95/98/NT. The files BLASTPOL.ZIP and MRGDPPOL.ZIP contain files necessary to perform polling using Blast Professional for DOS and Blast Data Pump, respectively.

Sample batch files for FTP polling are also included. They can be found in MRGFTP.ZIP. Files for using email polling are in the file MRGEMAIL.ZIP.

These are basic polling procedures similar to those used by most companies using MultiStore. You must set up subdirectory structures for **Keystroke** data files and enter appropriate telephone numbers, passwords, etc. Following are sample polling procedures with notes (text following **REM** or **;**) describing each step for use over a Local Area Network (LAN), for pcANYWHERE v9.2 for Windows, for email, and for FTP.

---

**NOTE:** *Since some command lines are longer than will fit on the printed page, some text is wrapped and indented to show that it is a continuation of the preceding line.*

---

# Sample Polling Script with 2 Stores, Both Live

```
REM ***** 2 LOCAL STORES *****
REM MERGENET.BAT - Storemerge Control File for 2 Stores &
REM Master Database at One Location.
REM Using LAN, WAN, or product like NT Terminal Server
REM          to make a real-time connection between all the stores.

ECHO OFF
BREAK ON

REM Backup/Delete previous merge files
F:
CD\KEYSTROK
DEL KSTRN001.MRG
CD\KEYSTROK\STORE1
..\DATENAME KSTRN001.MRG *.MRG
CD\KEYSTROK\STORE2
..\DATENAME KSTRN001.MRG *.MRG
CD\KEYSTROK
DEL *.ERR
DEL F:\KEYSTROK\STORE1\HOLD\KSTRN001.MRG
DEL F:\KEYSTROK\STORE2\HOLD\KSTRN001.MRG

REM Run UPDATE INVENTORY
STRUPDTE.EXE

REM Get Sales from STORE1
F:
CD\KEYSTROK
EXTRACT /D STORE1 /M STORE1\HOLD

REM Get Sales from STORE2
F:
CD\KEYSTROK
EXTRACT /D STORE2 /M STORE2\HOLD

REM Check for merge files (.mrg), make error files (.err)
REM          if needed.
F:
CD\KEYSTROK

REM Check for 0 Byte file by copying in DOS.
COPY F:\KEYSTROK\STORE1\HOLD\KSTRN001.MRG F:\KEYSTROK\STORE1\KSTRN001.MRG
COPY F:\KEYSTROK\STORE2\HOLD\KSTRN001.MRG F:\KEYSTROK\STORE2\KSTRN001.MRG
IF EXIST F:\KEYSTROK\STORE1\KSTRN001.MRG GOTO OK1
ECHO NO MERGE FILE>F:\KEYSTROK\STORE1.ERR
:OK1
IF EXIST F:\KEYSTROK\STORE2\KSTRN001.MRG GOTO OK2
ECHO NO MERGE FILE>F:\KEYSTROK\STORE2.ERR
:OK2
```

```
REM Merge STORE1 sales into Master
CD\KEYSTROK
IF EXIST F:\KEYSTROK\STORE1.ERR GOTO SKIP1
COPY F:\KEYSTROK\STORE1\KSTRN001.MRG
STOREMRG /NOP /NOINV
:SKIP1

REM Merge STORE2 sales into Master
IF EXIST F:\KEYSTROK\STORE2.ERR GOTO SKIP2
COPY F:\KEYSTROK\STORE2\KSTRN001.MRG
STOREMRG /NOP /NOINV
:SKIP2

REM Copy Customer Data files back to subdirs & reset customer next number.
IF EXIST F:\KEYSTROK\STORE1.ERR GOTO NOCUST1
COPY F:\KEYSTROK\DATA\KSCST*.DAT F:\KEYSTROK\STORE1\KSCST*.DAT
COPYNUM /D F:\KEYSTROK\STORE1 /M F:\KEYSTROK\STORE1
:NOCUST1

IF EXIST F:\KEYSTROK\STORE2.ERR GOTO NOCUST2
COPY F:\KEYSTROK\DATA\KSCST*.DAT F:\KEYSTROK\STORE2\KSCST*.DAT
COPYNUM /D F:\KEYSTROK\STORE2 /M F:\KEYSTROK\STORE2
:NOCUST2

@ECHO POLLING TRANSFER IS COMPLETE
```

# Sample Polling Script with a Remote Store

```
REM          ***** 1 REMOTE *****
REM MERGEAW1.BAT - Storemerge Control File for 1 Remote Store &
REM ***      Master Database and Local Store at a One Location. ***
REM          Using PCAnywhere for Windows95/98, version 9.2

@ECHO OFF
BREAK ON

REM Backup/Delete previous merge files
C:
CD\KEYSTROK
DEL KSTRN001.MRG
CD\KEYSTROK\STORE1
COPY KSTRN001.MRG *.BAK
DEL KSTRN001.MRG
CD\KEYSTROK\STORE2
COPY KSTRN001.MRG *.BAK
DEL KSTRN001.MRG
CD\KEYSTROK
DEL C:\KEYSTROK\STORE1\HOLD\KSTRN001.MRG
DEL C:\KEYSTROK\STORE2\HOLD\KSTRN001.MRG
DEL *.ERR

REM Run UPDATE INVENTORY
STRUPDTE.EXE

REM ***** NEW ***** Extract Purchase Orders from Store2 at the Master
REM          and send them to Store2
REM Un-REM lines below & in extract2.scr to activate.
REM del c:\keystrok\store2\hold\kstrn001.po
REM copy c:\keystrok\store2\extract.dat c:\keystrok\store2\extract.bak
REM EXTRACT.EXE /DSTORE2 EXTRACT=5 END=M /NOCUST
REM copy c:\keystrok\kstrn001.mrg c:\keystrok\store2\hold\kstrn001.po
REM STOREMRG.EXE /NOP /NOINV /NOCUST
REM DEL KSTRN001.MRG

REM Get Sales from STORE1
C:
CD\KEYSTROK
EXTRACT.EXE /D STORE1 /M STORE1\HOLD

REM Check for merge files (.mrg), make error files (.err) if needed.
REM Check for 0 Byte file by copying in DOS.
COPY C:\KEYSTROK\STORE1\HOLD\KSTRN001.MRG C:\KEYSTROK\STORE1\KSTRN001.MRG
IF EXIST C:\KEYSTROK\STORE1\KSTRN001.MRG GOTO OK1
ECHO NO MERGE FILE>C:\KEYSTROK\STORE1.ERR
:OK1
```

```

REM Merge STORE1 sales into Master
CD\KEYSTROK
IF EXIST C:\KEYSTROK\STORE1.ERR GOTO SKIP1
COPY C:\KEYSTROK\STORE1\KSTRN001.MRG
STOREMRG.EXE /NOP /NOINV
:SKIP1

REM Copy Customer Data files back to subdirs & reset customer next number.
IF EXIST C:\KEYSTROK\STORE1.ERR GOTO NOCUST1
COPY C:\KEYSTROK\DATA\KSCST*.DAT C:\KEYSTROK\STORE1\KSCST*.DAT
COPYNUM.EXE /D C:\KEYSTROK\STORE1 /M C:\KEYSTROK\STORE1
:NOCUST1

REM Get Sales from STORE2
C:
cd\progra~1\symantec\pcanyw~1\
START /W C:\progra~1\symantec\pcanyw~1\awrem32.exe
      C:\progra~1\symantec\pcanyw~1\data\extract2.scr
CLS

C:
CD\KEYSTROK
REM Check for merge files (.mrg), make error files (.err) if needed.
REM Check for 0 Byte file by copying in DOS.
COPY C:\KEYSTROK\STORE2\HOLD\KSTRN001.MRG C:\KEYSTROK\STORE2\KSTRN001.MRG
IF EXIST C:\KEYSTROK\STORE2\KSTRN001.MRG GOTO OK2
ECHO NO MERGE FILE>C:\KEYSTROK\STORE2.ERR
:OK2

REM Merge STORE2 sales into Master
IF EXIST C:\KEYSTROK\STORE2.ERR GOTO SKIP2
COPY C:\KEYSTROK\STORE2\KSTRN001.MRG
STOREMRG.EXE /NOP /NOINV

REM Merge STORE2 sales into STORE2 subdir
COPY C:\KEYSTROK\STORE2\KSTRN001.MRG
STOREMRG.EXE /NOP /D C:\KEYSTROK\STORE2
:SKIP2

REM Copy Customer Data files back to subdirs & reset customer next number.
IF EXIST C:\KEYSTROK\STORE2.ERR GOTO NOCUST2
COPY C:\KEYSTROK\DATA\KSCST*.DAT C:\KEYSTROK\STORE2\KSCST*.DAT
COPYNUM /D C:\KEYSTROK\STORE2 /M C:\KEYSTROK\STORE2
:NOCUST2

REM Create INVLINK files
INVLINK.EXE /D STORE1
INVLINK.EXE /D STORE2

```

```
REM Stamp Inventory & Customer files
STAMP.EXE STORE2\KSINV*.DAT
STAMP.EXE STORE2\KSCST*.DAT

REM Transfer Inventory, Customers, & InvLink files to STORE2 location.
IF EXIST C:\KEYSTROK\STORE2.ERR GOTO NOSEND2
C:
cd\progra~1\symantec\pcanyw~1\
START /W C:\progra~1\symantec\pcanyw~1\awrem32.exe
      C:\progra~1\symantec\pcanyw~1\data\send2.scr
:NOSEND2

REM Show Error (.err) files if they exist.
C:
CD\KEYSTROK
DIR *.ERR

@ECHO Polling is successful if "File not found" is printed above!
@ECHO
@ECHO POLLING TRANSFER IS COMPLETE
```

## Sample pcAnywhere Script Files

```
;Filename = extract2.scr      ***** pcANYWHERE32 Version 9.2 *****
Session ExitMode Accept      ;leaves host waiting to Accept another call.
Session OnError Ignore       ;Ignores error, continues to transfer files.
Session Overwrites Always
Set Timeout 1200              ;set timeout for inactivity, in seconds
On Timeout GoTo @timeout
Session Retry 1               ;number of retries to attempt
Session Delay 1               ;amount of time between retries in minutes
;Wait for 10 seconds for previous pcANYWHERE to close and reset modem.
Wait 10
;Dial Store#2 Computer
Session Dial "STORE2"
If $DCD != 1 GoTo @Timeout
;Send the file STORE2\HOLD\EXTRACT.DAT to remote location
SessOpr Remote Send "C:\KEYSTROK\STORE2\HOLD\EXTRACT.DAT"
        "C:\KEYSTROK\DATA\HOLD\EXTRACT.DAT"
;Run Extract and get merge file
SessOpr Host Run Wait "C:\KEYSTROK\EXT2.BAT"
;Wait for 10 seconds NOTE: May need to increase time
        to 60(1 mins) or 120(2 Mins)
Wait 10
SessOpr Host Send "C:\KEYSTROK\KSTRN001.MRG"
        "C:\KEYSTROK\STORE2\HOLD\KSTRN001.MRG"
SessOpr Host Send "C:\KEYSTROK\DATA\EXTRACT.DAT"
        "C:\KEYSTROK\STORE2\HOLD\EXTRACT.DAT"
;REM The lines below can be used to send Purchase Orders
        to the remote location.
;REM The file MRGPO2.BAT must be in the \KEYSTROK directory at the remote.
;SessOpr Remote Send "C:\KEYSTROK\STORE2\HOLD\KSTRN001.PO"
        "C:\KEYSTROK\DATA\HOLD\KSTRN001.PO"
;SessOpr Host Run Wait "C:\KEYSTROK\MRGPO2.BAT"
;REM Wait for 20 seconds
;Wait 20
@timeout:
Exit
```

```

;Filename = send2.scr          ***** pcANYWHERE32 Version 9.2 *****
Session ExitMode Accept      ;leaves host waiting to Accept another call.
Session OnError Ignore       ;Ignores error, continues to transfer files.
Session Overwrites Always
Set Timeout 1200             ;set timeout for inactivity, in seconds
On Timeout GoTo @timeout
Session Retry 1              ;number of retries to attempt
Session Delay 1              ;amount of time between retries in minutes
;Wait for 10 seconds for previous pcANYWHERE to close.
Wait 10
;Dial Store#2 Location.
Session Dial "STORE2"
If $DCD != 1 GoTo @timeout
;Delete OLD Inventory & Customers from STORE2 directory at STORE2 loca-
tion.
SessOpr Host Run Wait "DEL C:\KEYSTROK\DATA\HOLD\KSINV*.DAT"
SessOpr Host Run Wait "DEL C:\KEYSTROK\DATA\HOLD\KSCST*.DAT"
;Send NEW Inventory files to STORE2 directory at STORE2 location.
SessOpr Remote Send "C:\KEYSTROK\STORE2\KSINV.DAT"
"C:\KEYSTROK\DATA\HOLD\KSINV.DAT"
SessOpr Remote Send "C:\KEYSTROK\STORE2\KSINV1.DAT"
"C:\KEYSTROK\DATA\HOLD\KSINV1.DAT"
SessOpr Remote Send "C:\KEYSTROK\STORE2\KSINV2.DAT"
"C:\KEYSTROK\DATA\HOLD\KSINV2.DAT"
SessOpr Remote Send "C:\KEYSTROK\STORE2\KSINVNDX.DAT"
"C:\KEYSTROK\DATA\HOLD\KSINVNDX.DAT"
SessOpr Remote Send "C:\KEYSTROK\STORE2\KSINVCMT.DAT"
"C:\KEYSTROK\DATA\HOLD\KSINVCMT.DAT"
SessOpr Remote Send "C:\KEYSTROK\STORE2\KSINVSER.DAT"
"C:\KEYSTROK\DATA\HOLD\KSINVSER.DAT"
;Send Customers files to STORE2 directory at STORE2 location.
SessOpr Remote Send "C:\KEYSTROK\STORE2\KSCST.DAT"
"C:\KEYSTROK\DATA\HOLD\KSCST.DAT"
SessOpr Remote Send "C:\KEYSTROK\STORE2\KSCST1.DAT"
"C:\KEYSTROK\DATA\HOLD\KSCST1.DAT"
SessOpr Remote Send "C:\KEYSTROK\STORE2\KSCST2.DAT"
"C:\KEYSTROK\DATA\HOLD\KSCST2.DAT"
SessOpr Remote Send "C:\KEYSTROK\STORE2\KSCSTNDX.DAT"
"C:\KEYSTROK\DATA\HOLD\KSCSTNDX.DAT"
SessOpr Remote Send "C:\KEYSTROK\STORE2\KSCSTCMT.DAT"
"C:\KEYSTROK\DATA\HOLD\KSCSTCMT.DAT"
;Send Inventory Link file to KEYSTROK directory at STORE2 location.
SessOpr Remote Send "C:\KEYSTROK\KSIL2.DAT" "C:\KEYSTROK\DATA\KSIL2.DAT"
;Run CHKSTAMP.BAT at STORE2 location to verify Successful Transfers.
SessOpr Host Run Wait "C:\KEYSTROK\CHKSTAMP.BAT"
;Wait to copy files. If large files or slow computer, increase time.
Wait 10
@timeout:
Exit

```

## MultiStore Polling via Email

Because many companies today have internet and email connections, you may want to consider automating the polling process to transfer data via email attachments. By using standard email accounts, you can replace the need for a dial up communications package such as pcAnywhere. Two **Keystroke POS** add-on utilities, **GETMAIL.EXE** and **SENDMAIL.EXE** can be used to control how emails are sent. These two utilities are included with the **Keystroke** User Bundle; for more information on **GETMAIL.EXE** or **SENDMAIL.EXE**, please refer to the *Windows Features and Version 4.1 Addendum*.

A typical polling process conducted via email would include the following 3 steps:

1. Sending of transaction data from each Remote location to the Master location.
2. Processing of the transaction data and merging transactions into a consolidated database.
3. Sending updated Inventory (and other records) back to each Remote location.

### *Sending transaction data from a Remote location to the Master location*

Each store will need to run a batch program through a task scheduler such as Microsoft Windows98 Task Scheduler (Start – Programs – Accessories – System Tools – Scheduled Tasks). The batch file will need to run **EXTRACT.EXE** and **SENDMAIL.EXE**, among other commands, to send new transactions and customers to the Master location. Following is an example batch file (**EXT1.BAT**) for Store1:

```
REM FILENAME=EXT1.BAT
REM Batch file to be run at Store 1 on a scheduled basis.
REM This batch file will extract current transaction information
REM and email the resulting merge file (MERGE1.ZIP)
REM This can be run multiple times a day, as needed.
C:
CD\KEYSTROK
DEL KSTRN001.MRG
REM Copy EXTRACT.DAT file to *.BAK
COPY C:\KEYSTROK\DATA\EXTRACT.DAT C:\KEYSTROK\DATA\EXTRACT.BAK
REM Extract transaction data from Store 1.
REM This will extract transactions up through 30 minutes ago.
REM This is in case computer clocks are off over a network
REM within 30 minutes
EXTRACT.EXE ENDTIME=-30M
REM Make the file MERGE1.ZIP (using a password)
REM to be sent to Master Location.
DEL MERGE1.ZIP
PKZIP MERGE1.ZIP KSTRN001.MRG -SPASSWORD
REM Use DATENAME.EXE to rename KSTRN001.MRG to {DATE}.DAT
REM This file is placed in the C:\KEYSTROK\DATA directory.
DATENAME.EXE KSTRN001.MRG DATA\*.DAT
REM Send an Email with attached file to Master Location.
START /W SENDMAIL.EXE INIFILE=EXT1.INI
REM END
```

This batch file creates the file **KSTRN001.MRG** by running **EXTRACT.EXE**, then compresses (zips) the file with a password, creating the file **MERGE1.ZIP**. It then renames the **.MRG** file to **DATA\{Today'sDate}.DAT** and sends an email with the attachment by using the file **EXT1.INI**. Below is an example **EXT1.INI**:

```
UserName=MS Exchange Settings
Password=
SendTo=MasterEmailAddress@YourEmailProvider.com
Subject=Extract Data from Store 1
NoteText=Attached file is a zipped up merge file from Store 1
Attachment=C:\KEYSTROK\MERGE1.ZIP
```

This INI file will use the Microsoft Outlook profile and create an email with an attached merge file. Your email program must be set up to automatically send/receive emails.

---

*NOTE: Upper and Lowercase is important within the INI file.*

---

## ***Processing transaction data and merging transactions into a consolidated database***

Below is an example batch file (**MRGSALES.BAT**) for retrieving the email message from **Store1** and **Store2**, then merging transactions and customers:

```
@ECHO OFF
REM FILENAME=MRGSALES.BAT
REM Batch file to process merge files received via Email.
REM To be run from Master Location.
C:
CD\KEYSTROK
:STORE1
REM Get Email from Store1
C:
CD\KEYSTROK
START /W GETMAIL.EXE INIFILE=GETSALE1.INI NoPause=1
REM Process merge file from Store 1
C:
CD\KEYSTROK\STORE1
IF EXIST MERGE1.ZIP GOTO PROCESS1
REM No file to process, skip to NEXT STORE
ECHO NO MERGE FILE>C:\KEYSTROK\STORE1.ERR
GOTO STORE2
:PROCESS1
CD\KEYSTROK
PKUNZIP STORE1\MERGE1.ZIP STORE1 -sPASSWORD -o
REM Rename MERGE1.ZIP so it doesn't get processed again,
REM and since GETMAIL won't save a file if it already exists.
CD\KEYSTROK
DATENAME.EXE STORE1\MERGE1.ZIP STORE1\*.ZIP
IF EXIST C:\KEYSTROK\STORE1\KSTRN001.MRG GOTO OK1
ECHO NO MERGE FILE>C:\KEYSTROK\STORE1.ERR
:OK1
CD\KEYSTROK
```

```

IF EXIST STORE1.ERR GOTO STORE2
DEL KSTRN001.MRG
REM Merge sales into Master directory.
COPY STORE1\KSTRN001.MRG
STOREMRG.EXE /NOP /NOINV
REM Merge sales into Store 1 directory.
COPY STORE1\KSTRN001.MRG
STOREMRG.EXE /DSTORE1 /NOP
REM Check for another Email from Store 1.
GOTO STORE1
:STORE2
REM Get Email from Store2
C:
CD\KEYSTROK
START /W GETMAIL.EXE INIFILE=GETSALE2.INI NoPause=1
REM Process merge file from Store 2
C:
CD\KEYSTROK\STORE2
IF EXIST MERGE2.ZIP GOTO PROCESS2
REM No file to process, skip to END
ECHO NO MERGE FILE>C:\KEYSTROK\STORE2.ERR
GOTO END
:PROCESS2
CD\KEYSTROK
PKUNZIP STORE2\MERGE2.ZIP STORE2 -sPASSWORD -o
IF EXIST C:\KEYSTROK\STORE2\KSTRN001.MRG GOTO OK2
ECHO NO MERGE FILE>C:\KEYSTROK\STORE2.ERR
:OK2
REM Rename MERGE2.ZIP so it doesn't get processed again,
REM and since GETMAIL won't save a file if it already exists.
CD\KEYSTROK
DATENAME.EXE STORE2\MERGE2.ZIP STORE2\*.ZIP
CD\KEYSTROK
IF EXIST STORE2.ERR GOTO END
DEL KSTRN001.MRG
REM Merge sales into Master directory.
COPY STORE2\KSTRN001.MRG
STOREMRG.EXE /NOP /NOINV
REM Merge sales into Store 2 directory.
COPY STORE2\KSTRN001.MRG
STOREMRG.EXE /DSTORE2 /NOP
REM Check for another Email from Store 2.
GOTO STORE2
:END
C:
CD\KEYSTROK
CLS
DIR *.ERR
ECHO      * * * * POLLING COMPLETE * * * *

```

Below is an example of **GETSALE1.INI** used with **GETMAIL.EXE** in the above **MRGSALES.BAT** file.

```
UserName=MS Exchange Settings
Password=
Subject=Extract Data from Store 1
NoteText=Attached file is a zipped up merge file from Store 1
AddSubjectToPath=0
SaveTo=C:\KEYSTROK\STORE1\MERGE1.ZIP
NoPause=1
```

### *Sending updated Inventory back to each Remote location*

Below is an example batch file (**SENDINV1.BAT**) for sending the updated Inventory back to a Remote location (**Store1**):

```
REM FILENAME=SENDINV1.BAT
REM Batch file to be run at Master on a scheduled basis.
REM This batch file will send updated Inventory information to Store 1
REM The file sent is INV1.ZIP
REM **** THIS SHOULD ONLY BE RUN WHEN STORE1 IS NOT IN KEYSTROKE POS ****
C:
CD\KEYSTROK
IF EXIST STORE1.ERR GOTO END
REM Run Update Inventory
STRUPDTE.EXE
REM Stamp files to be transferred to Store1.
STAMP.EXE STORE1\KSINV*.DAT
STAMP.EXE STORE1\KSCST*.DAT
STAMP.EXE STORE1\KSDPT*.DAT
REM STAMP.EXE STORE1\KSVND*.DAT
REM STAMP.EXE STORE1\KSCLK*.DAT
REM Copy files to be transferred to the STORE1\SEND directory.
COPY C:\KEYSTROK\STORE1\KSINV*.DAT C:\KEYSTROK\STORE1\SEND\KSINV*.DAT
COPY C:\KEYSTROK\STORE1\KSCST*.DAT C:\KEYSTROK\STORE1\SEND\KSCST*.DAT
COPY C:\KEYSTROK\STORE1\KSDPT*.DAT C:\KEYSTROK\STORE1\SEND\KSDPT*.DAT
REM COPY C:\KEYSTROK\STORE1\KSVND*.DAT C:\KEYSTROK\STORE1\SEND\KSVND*.DAT
REM COPY C:\KEYSTROK\STORE1\KSCLK*.DAT C:\KEYSTROK\STORE1\SEND\KSCLK*.DAT
REM Make the file INV1.ZIP to be sent to Store 1 Location.
DEL INV1.ZIP
PKZIP.EXE INV1.ZIP STORE1\SEND\*.DAT -sPASSWORD
REM Send an Email with attached file to Store 1 Location.
START /W SENDMAIL.EXE INIFILE=SENDINV1.INI
:END
CLS
DIR STORE1.ERR
```

Following is an example of the file **SENDINV1.INI** used in the above **SENDINV1.BAT** file to send Inventory to Store1:

```
UserName=MS Exchange Settings
Password=
SendTo=MasterEmailAddress@YourEmailProvider.com
Subject=Inventory files for Store 1
NoteText=Attached file is a zipped up file for Store 1
Attachment=C:\KEYSTROK\INV1.ZIP
```

An additional batch file (**GETINV1.BAT**) must also be scheduled at the Remote location to unzip and replace the files received from the Master:

```
REM FILENAME=GETINV1.BAT
REM Batch file to process updated Inventory & Customers received via
Email.
REM To be run from Store1 Location.
REM **** EVERYONE SHOULD BE OUT OF THE KEYSTROKE PROGRAM ****
REM This batch file should only be run after business hours.
C:
CD\KEYSTROK
:START
REM Get Email from Master
START /W GETMAIL.EXE INIFILE=GETINV1.INI NoPause=1
REM Process INV1.ZIP file(s) if they exist
C:
CD\KEYSTROK
IF EXIST INV1.ZIP GOTO PROCESS1
REM No file to process, skip to END
GOTO END
:PROCESS1
REM First the new sales and customers will be extracted and sent to Master
REM Copy EXTRACT.DAT file to *.BAK
COPY C:\KEYSTROK\DATA\EXTRACT.DAT C:\KEYSTROK\DATA\EXTRACT.BAK
REM Extract transaction data from Store 1.
REM This will extract transactions up through 10 minutes ago.
REM in case a previous extract did not complete.
EXTRACT.EXE ENDTIME=-10M
REM Make the file MERGE1.ZIP to be sent to Master Location.
DEL MERGE1.ZIP
PKZIP MERGE1.ZIP KSTRN001.MRG -SPASSWORD
REM Use DATANAME.EXE to rename KSTRN001.MRG to {DATE}.DAT
REM This file is placed in the C:\KEYSTROK\DATA directory.
DATANAME.EXE KSTRN001.MRG DATA\*.DAT
REM Send an Email with attached file to Master Location.
START /W SENDMAIL.EXE INIFILE=EXT1.INI
REM **** Inventory & Customers will be copied over live data. ****
REM **** EVERYONE SHOULD BE OUT OF THE KEYSTROKE PROGRAM ****
CD\KEYSTROK
PKUNZIP.EXE INV1.ZIP DATA\HOLD -SPASSWORD -o
REM Rename INV1.ZIP so it doesn't get processed again,
REM and since GETMAIL won't save a file if it already exists.
CD\KEYSTROK
DATANAME.EXE INV1.ZIP DATA\HOLD\*.ZIP
```

```

REM Verify files were received correctly and copy over live data files.
COPY C:\KEYSTROK\DATA\KSINV*.DAT C:\KEYSTROK\DATA\KSINV*.BAK
COPY C:\KEYSTROK\DATA\KSCST*.DAT C:\KEYSTROK\DATA\KSCST*.BAK
CALL CHKSTAMP.BAT
REM Check for another Email from Master.
GOTO START
:END
C:
CD\KEYSTROK

```

Below is an example of the file **GETINV1.INI** used above with **GETMAIL.EXE** to receive Inventory at Store1:

```

UserName=MS Exchange Settings
Password=
Subject=Inventory files for Store 1
NoteText=Attached file is a zipped up file for Store 1
AddSubjectToPath=0
SaveTo=C:\KEYSTROK\INV1.ZIP
NoPause=1

```

## ***Files Needed for Polling***

Below is a listing of files required for Automated Polling. All of the below files are included in the file **MRGEMAIL.ZIP**. These files are sorted based on their function within the polling process.

### ***Extract and send transaction data to the Master location via Email, sending the files MERGE1.ZIP and/or MERGE2.ZIP***

**EXT1.BAT** – This file is used on a scheduler to extract transaction data and Email the zipped up merge file. Place this file in the \KEYSTROK directory at Store1 remote.

**EXT1.INI** – This file used by EXT1.BAT to specify how to send the Email. You will need to modify this file to include User Name, Password (if used in your email profile), and Email Address. Place this file in the \KEYSTROK directory at Store1 remote.

**EXT2.BAT** – This file is used on a scheduler to extract transaction data and Email the zipped up merge file. Place this file in the \KEYSTROK directory at Store2 remote.

**EXT2.INI** – This file used by EXT2.BAT to specify how to send the Email. You will need to modify this file to include User Name, Password (if used in your email profile), and Email Address. Place this file in the \KEYSTROK directory at Store2 remote.

***Get the transaction data (MERGE1.ZIP and MERGE2.ZIP) from the Email Program and merge it into the Consolidated database and the subdirectories at the Master location.***

**MRGSALES.BAT** – This file is used on a scheduler to check for Emails and processes them.  
NOTE: This is written for 2 REMOTE locations. You will need to edit this file if one store is at the master. Place this file in the \KEYSTROK directory at the Master.

**GETSALE1.INI** – File used by MRGSALES.BAT to specify how to receive the Email. You will need to modify this file to include User Name, Password (if used in your email profile), and Email Address. Place this file in the \KEYSTROK directory at the Master.

**GETSALE2.INI** – File used by MRGSALES.BAT to specify how to receive the Email. You will need to modify this file to include User Name, Password (if used in your email profile), and Email Address. Place this file in the \KEYSTROK directory at the Master.

***Send updated Inventory & Customers to Remote locations after verifying the merge file has been received.***

**SENDINV.BAT** – File used to send Inventory files to both Store1 & Store2. Place this file in the \KEYSTROK directory at the Master.

**SENDINV1.BAT** – File used to send Inventory files to Store1 only. Place this file in the \KEYSTROK directory at the Master.

**SENDINV1.INI** – File used by SENDINV1.BAT to specify how to send the Email. Place this file in the \KEYSTROK directory at the Master.

**SENDINV2.BAT** – File used to send Inventory files to Store2 only. Place this file in the \KEYSTROK directory at the Master.

**SENDINV2.INI** – File used by SENDINV2.BAT to specify how to send the Email. Place this file in the \KEYSTROK directory at the Master.

***Get updated Inventory & Customers to replace Remote data.***

**GETINV1.BAT** – File used to process Inventory files received from the Master. Place this file in the \KEYSTROK directory at Store1.

**GETINV1.INI** – File used by GETINV1.BAT to specify how to receive the Email. Place this file in the \KEYSTROK directory at Store1.

**GETINV2.BAT** – File used to process Inventory files received from the Master. Place this file in the \KEYSTROK directory at Store2.

**GETINV2.INI** – File used by GETINV2.BAT to specify how to receive the Email. Place this file in the \KEYSTROK directory at Store2.

## ***Suggested Schedule of Events***

Following is a suggested schedule for nightly polling.

- EXT1.BAT** (scheduled to run at 10:30p.m. each night at Store1)
- EXT2.BAT** (scheduled to run at 10:30p.m. each night at Store2)
- MRGSALES.BAT** (scheduled to run at 11:00p.m. each night at the Master)
- SENDINV.BAT** (scheduled to run at 11:30p.m. each night at the Master)
- GETINV1.BAT** (scheduled to run at 12:00a.m. each night at Store1)
- GETINV2.BAT** (scheduled to run at 12:00a.m. each night at Store2)
- MRGSALES.BAT** (scheduled to run at 12:30a.m. each night at the Master)

## ***Directory Structure***

The following directory structure is required in order for these sample polling scripts to operate correctly. Please refer to the Data Configurations section of this Appendix for detailed information on setting up MultiStore and preparing the data files for polling.

---

***NOTE:** If you are running **Keystroke POS** over a network, the drive letters in these polling scripts will need to be changed in some places.*

---

In addition to the directories defined earlier in this Appendix, the following directories will also need to be created:

<b>C:\KEYSTROK\STORE1\SEND</b>	<b>at Master</b>
<b>C:\KEYSTROK\STORE2\SEND</b>	<b>at Master</b>
<b>C:\KEYSTROK\DATA\HOLD</b>	<b>at Store1</b>
<b>C:\KEYSTROK\DATA\HOLD</b>	<b>at Store2</b>

## ***MultiStore Polling via FTP***

An FTP (File Transfer Protocol) site may also be used to transfer data between store locations. In the examples below, it is assumed that the one of the stores will be the Master location and maintain a set of databases. These include one for itself (**STORE1\**), a consolidated database (**DATA\**), and a database that mimics a remote location (**STORE2\**).

New transactions and customers added at the remote (Store 2) site will be sent to the Master location via the FTP site, and updated customer and inventory records will be sent back to the remote (Store 2) site. Other databases that are maintained at the Master location can also be sent to the remote site (Vendors, Departments, Clerks, Price Tables, etc.).

A typical polling process would include the following steps:

1. The Remote location schedules a batch file to run Extract and create the file to be sent to the FTP site.
2. With a FTP program, the Remote location schedules the uploading of the file just created to a directory on the FTP site.
3. With a FTP program, the Master location schedules the downloading of the same file from the FTP site.
4. The Master location schedules a batch file to process the received file(s) and prepare a package of 'new files' to be returned to the Remote location.
5. The Master location schedules the uploading of the 'new files' to a directory on the FTP site.
6. The Remote location schedules the downloading of the file(s) from the FTP site.
7. The Remote location schedules a batch file to process the files received.

### ***Sending transaction data from a Remote location to the FTP site***

Each store will need to run a batch program through a task scheduler like Microsoft Windows98 Task Scheduler (Start – Programs – Accessories – System Tools – Scheduled Tasks). The batch file will need to run **EXTRACT.EXE** and **PKZIP.EXE**, among other commands, to send new transactions and customers to the FTP site. Following is an example batch file for Store 1:

```
@ECHO OFF
REM FILENAME=EXTFTP2.BAT
REM Batch file to be run at Store 2
REM This batch file will extract current transaction information
REM and create the file MERGE2.ZIP.
REM This file will be sent to the main office via FTP Program
REM 11/30/01 - Tested with FTP Voyager 9.0.0.4 by www.Rhinosoft.com
```

```

REM Prepare for extract
C:
CD\KEYSTROK
DEL C:\KEYSTROK\MERGE2.ZIP
DEL C:\KEYSTROK\DATA\HOLD\*.DAT
DEL C:\KEYSTROK\DATA\HOLD\*.ZIP
DEL C:\KEYSTROK\KSTRN001.MRG

REM Copy EXTRACT.DAT file to *.BAK
COPY C:\KEYSTROK\DATA\EXTRACT.DAT C:\KEYSTROK\DATA\EXTRACT.BAK

REM Extract transaction data from C:\KEYSTROK\DATA
EXTRACT.EXE

REM Makes the file MERGE2.ZIP to be sent to Master Location.
COPY C:\KEYSTROK\KSTRN001.MRG C:\KEYSTROK\DATA\HOLD\KSTRN001.MRG
COPY C:\KEYSTROK\DATA\EXTRACT.DAT C:\KEYSTROK\DATA\HOLD\EXTRACT.DAT
PKZIP.EXE MERGE2.ZIP C:\KEYSTROK\DATA\HOLD\*.*

REM Use DATENAME.EXE to rename KSTRN001.MRG to {DATE}.DAT
REM This file is placed in the C:\KEYSTROK\DATA directory.
DATENAME.EXE KSTRN001.MRG C:\KEYSTROK\DATA\*.DAT

```

This batch file creates the file **KSTRN001.MRG** by running **EXTRACT.EXE**, then compresses (zips) the file along with **EXTRACT.DAT** in the file **MERGE2.ZIP**. It then renames the **.MRG** file to **C:\KEYSTROK\DATA\{Today'sDate}.DAT**.

### ***Processing transaction data and merging transactions into a consolidated database***

Below is an example of a batch file which will process transaction data, then prepare files to be sent via FTP. The file to be sent via FTP is **SEND2.ZIP** from the **C:\KEYSTROK\STORE2\SEND** directory.

```

@ECHO OFF
REM MRGFTP1.BAT - Storemerge Control File for 1 Remote Store .
REM Master Database and Local Store are at a One Location.
REM Using FTP to transfer files.
REM Tested with FTP Voyager by www.RhinoSoft.com
REM Local Store (STORE1) will maintain these databases
REM INVENTORY
REM DEPARTMENTS
REM VENDORS
REM CLERKS
REM SECURITY LEVELS
REM The Master Database (DATA) will maintain these databases
REM CUSTOMERS
REM ACCOUNTS RECEIVABLES

```

```

REM Backup/Delete previous merge files
C:
CD\KEYSTROK
DEL KSTRN001.MRG
CD\KEYSTROK\STORE1
COPY KSTRN001.MRG *.BAK
DEL KSTRN001.MRG
CD\KEYSTROK\STORE2
COPY KSTRN001.MRG *.BAK
DEL KSTRN001.MRG
CD\KEYSTROK
DEL *.ERR
DEL C:\KEYSTROK\STORE1\HOLD\KSTRN001.MRG
DEL C:\KEYSTROK\STORE2\HOLD\KSTRN001.MRG
DEL C:\KEYSTROK\STORE2\SEND\*.DAT
DEL C:\KEYSTROK\STORE2\SEND\*.ZIP

REM Run UPDATE INVENTORY
CD\KEYSTROK
STRUPDTE.EXE

REM Get Sales from STORE1
C:
CD\KEYSTROK
EXTRACT /D STORE1 /M STORE1\HOLD

REM Check for 0 Byte file by copying in DOS.
COPY C:\KEYSTROK\STORE1\HOLD\KSTRN001.MRG
      C:\KEYSTROK\STORE1\KSTRN001.MRG
IF EXIST C:\KEYSTROK\STORE1\KSTRN001.MRG GOTO OK1
ECHO NO MERGE FILE>C:\KEYSTROK\STORE1.ERR
:OK1

REM Merge STORE1 sales into Master
CD\KEYSTROK
IF EXIST C:\KEYSTROK\STORE1.ERR GOTO SKIP1
COPY C:\KEYSTROK\STORE1\KSTRN001.MRG
STOREMRG /NOP /NOINV
:SKIP1

REM Get Sales from STORE2

REM Unzip the files transferred by FTP
CD\KEYSTROK\STORE2\HOLD
C:\KEYSTROK\PKUNZIP.EXE C:\KEYSTROK\STORE2\HOLD\MERGE2.ZIP -O
DEL C:\KEYSTROK\STORE2\HOLD\MERGE2.ZIP

```

```

REM Check for merge files (.mrg), make error files (.err) if needed.
C:
CD\KEYSTROK
COPY C:\KEYSTROK\STORE2\HOLD\KSTRN001.MRG
      C:\KEYSTROK\STORE2\KSTRN001.MRG
IF EXIST C:\KEYSTROK\STORE2\KSTRN001.MRG GOTO OK2
ECHO NO MERGE FILE>C:\KEYSTROK\STORE2.ERR
:OK2

REM Merge STORE2 sales into Master
IF EXIST C:\KEYSTROK\STORE2.ERR GOTO SKIP2
COPY C:\KEYSTROK\STORE2\KSTRN001.MRG
STOREMRG /NOP /NOINV

REM Merge STORE2 sales into STORE2 subdir
COPY C:\KEYSTROK\STORE2\KSTRN001.MRG
STOREMRG /NOP /NOCUST /D C:\KEYSTROK\STORE2
:SKIP2

REM Copy Customer Data files back to subdirs & reset customer next number.
IF EXIST C:\KEYSTROK\STORE1.ERR GOTO NOCUST1
COPY C:\KEYSTROK\DATA\KSCST*.DAT C:\KEYSTROK\STORE1\KSCST*.DAT
COPYNUM /D C:\KEYSTROK\STORE1 /M C:\KEYSTROK\STORE1
:NOCUST1
IF EXIST C:\KEYSTROK\STORE2.ERR GOTO NOCUST2
COPY C:\KEYSTROK\DATA\KSCST*.DAT C:\KEYSTROK\STORE2\KSCST*.DAT
COPYNUM /D C:\KEYSTROK\STORE2 /M C:\KEYSTROK\STORE2
:NOCUST2

REM Create INVLINK files
INVLINK /D STORE1
INVLINK /D STORE2

REM Copy other DATABASES from STORE1 to other directories (optional)
REM COPY C:\KEYSTROK\STORE1\KSCLK*.DAT C:\KEYSTROK\DATA\KSCLK*.DAT
REM COPY C:\KEYSTROK\STORE1\LEVELS.DAT C:\KEYSTROK\DATA\LEVELS.DAT
REM COPY C:\KEYSTROK\STORE1\KSDPT*.DAT C:\KEYSTROK\DATA\KSDPT*.DAT
REM COPY C:\KEYSTROK\STORE1\ALTIC*.DAT C:\KEYSTROK\DATA\ALTIC*.DAT
REM COPY C:\KEYSTROK\STORE1\KSVND*.DAT C:\KEYSTROK\DATA\KSVND*.DAT
REM COPY C:\KEYSTROK\STORE1\KSCLK*.DAT C:\KEYSTROK\STORE2\KSCLK*.DAT
REM COPY C:\KEYSTROK\STORE1\LEVELS.DAT C:\KEYSTROK\STORE2\LEVELS.DAT
REM COPY C:\KEYSTROK\STORE1\KSDPT*.DAT C:\KEYSTROK\STORE2\KSDPT*.DAT
REM COPY C:\KEYSTROK\STORE1\ALTIC*.DAT C:\KEYSTROK\STORE2\ALTIC*.DAT
REM COPY C:\KEYSTROK\STORE1\KSVND*.DAT C:\KEYSTROK\STORE2\KSVND*.DAT

REM Stamp Inventory & Customer files (and others).
STAMP STORE2\KSINV*.DAT
STAMP STORE2\KSCST*.DAT
REM STAMP STORE2\KSDPT*.DAT
REM STAMP STORE2\KSCLK*.DAT
REM STAMP STORE2\KSVND*.DAT

```

```

REM Transfer Inventory, Customers, & InvLink files to STORE2 location.
IF EXIST C:\KEYSTROK\STORE2.ERR GOTO NOSEND2
REM COPY DATA FILES TO STORE2\SEND DIRECTORY
COPY C:\KEYSTROK\STORE2\KSINV*.DAT C:\KEYSTROK\STORE2\SEND\KSINV*.DAT
COPY C:\KEYSTROK\STORE2\KSCST*.DAT C:\KEYSTROK\STORE2\SEND\KSCST*.DAT
COPY C:\KEYSTROK\KSIL1.DAT C:\KEYSTROK\STORE2\SEND\KSIL1.DAT
REM COPY C:\KEYSTROK\STORE2\KSDPT*.DAT C:\KEYSTROK\STORE2\SEND\KSDPT*.DAT
REM COPY C:\KEYSTROK\STORE2\KSCLK*.DAT C:\KEYSTROK\STORE2\SEND\KSCLK*.DAT
REM COPY C:\KEYSTROK\STORE2\KSVND*.DAT C:\KEYSTROK\STORE2\SEND\KSVND*.DAT
GOTO SENDOK2
:NOSEND2
COPY C:\KEYSTROK\STORE2\HOLD\EXTRACT.DAT
C:\KEYSTROK\STORE2\SEND\EXTRACT.DAT
:SENDOK2

REM ZIP-UP all files in the STORE2\SEND directory
CD\KEYSTROK\
C:\KEYSTROK\PKZIP.EXE SEND2.ZIP C:\KEYSTROK\STORE2\SEND\*. *
REM The file C:\KEYSTROK\SEND2.ZIP will be transferred via FTP
REM to the remote location C:\KEYSTROK\DATA\HOLD and unzipped.

REM Show Error (.err) files if they exist.
C:
CD\KEYSTROK
DIR *.ERR

@ECHO Polling is successful if "File not found" is printed above!
@ECHO
@ECHO POLLING TRANSFER IS COMPLETE

```

## *Sending updated Inventory back to the Remote location*

Below is an example of a batch file (**CHKFTP.BAT**) which will decompress (unzip) the file **SEND2.ZIP**. This batch file is then used to unzip and copy transferred files back to the Remote location.

```

@ECHO OFF
REM File CHKFTP.BAT
REM Runs CHKSTAMP.EXE to verify files were transferred Ok.
REM Place this file in the KEYSTROK\ directory at each Remote location.

DEL C:\KEYSTROK\DATA\HOLD\*.DAT
IF NOT EXIST C:\KEYSTROK\DATA\HOLD\SEND2.ZIP GOTO NOFILE
C:
CD\KEYSTROK\DATA\HOLD
C:\KEYSTROK\PKUNZIP.EXE C:\KEYSTROK\DATA\HOLD\SEND2.ZIP -O

CD\KEYSTROK
IF EXIST C:\KEYSTROK\DATA\HOLD\EXTRACT.DAT GOTO EXTRACT

```

```

IF NOT EXIST C:\KEYSTROK\DATA\HOLD\KSINV.DAT GOTO MISSING1
IF NOT EXIST C:\KEYSTROK\DATA\HOLD\KSINV1.DAT GOTO MISSING1
IF NOT EXIST C:\KEYSTROK\DATA\HOLD\KSINV2.DAT GOTO MISSING1
IF NOT EXIST C:\KEYSTROK\DATA\HOLD\KSINVNDX.DAT GOTO MISSING1
IF NOT EXIST C:\KEYSTROK\DATA\HOLD\KSINVCMT.DAT GOTO MISSING1
IF NOT EXIST C:\KEYSTROK\DATA\HOLD\KSINVSER.DAT GOTO MISSING1
CHKSTAMP.EXE DATA\HOLD\KSINV*.DAT
IF ERRORLEVEL=1 GOTO BADINV
COPY C:\KEYSTROK\DATA\HOLD\KSINV*.DAT C:\KEYSTROK\DATA\KSINV*.DAT
:BADINV
:MISSING1
IF NOT EXIST C:\KEYSTROK\DATA\HOLD\KSCST.DAT GOTO MISSING2
IF NOT EXIST C:\KEYSTROK\DATA\HOLD\KSCST1.DAT GOTO MISSING2
IF NOT EXIST C:\KEYSTROK\DATA\HOLD\KSCST2.DAT GOTO MISSING2
IF NOT EXIST C:\KEYSTROK\DATA\HOLD\KSCSTNDX.DAT GOTO MISSING2
IF NOT EXIST C:\KEYSTROK\DATA\HOLD\KSCSTCMT.DAT GOTO MISSING2
CHKSTAMP.EXE DATA\HOLD\KSCST*.DAT
IF ERRORLEVEL=1 GOTO BADCUST
COPY C:\KEYSTROK\DATA\HOLD\KSCST*.DAT C:\KEYSTROK\DATA\KSCST*.DAT
:BADCUST
:MISSING2
IF NOT EXIST C:\KEYSTROK\DATA\HOLD\KSVND.DAT GOTO MISSING3
IF NOT EXIST C:\KEYSTROK\DATA\HOLD\KSVND1.DAT GOTO MISSING3
IF NOT EXIST C:\KEYSTROK\DATA\HOLD\KSVND2.DAT GOTO MISSING3
IF NOT EXIST C:\KEYSTROK\DATA\HOLD\KSVNDNDX.DAT GOTO MISSING3
IF NOT EXIST C:\KEYSTROK\DATA\HOLD\KSVNDCMT.DAT GOTO MISSING3
CHKSTAMP.EXE DATA\HOLD\KSVND*.DAT
IF ERRORLEVEL=1 GOTO BADVEND
COPY C:\KEYSTROK\DATA\HOLD\KSVND*.DAT C:\KEYSTROK\DATA\KSVND*.DAT
:BADVEND
:MISSING3
IF NOT EXIST C:\KEYSTROK\DATA\HOLD\KSDPT.DAT GOTO MISSING4
IF NOT EXIST C:\KEYSTROK\DATA\HOLD\KSDPTNDX.DAT GOTO MISSING4
IF NOT EXIST C:\KEYSTROK\DATA\HOLD\KSDPTCMT.DAT GOTO MISSING4
CHKSTAMP.EXE DATA\HOLD\KSDPT*.DAT
IF ERRORLEVEL=1 GOTO BADDEPT
COPY C:\KEYSTROK\DATA\HOLD\KSDPT*.DAT C:\KEYSTROK\DATA\KSDPT*.DAT
:BADDEPT
:MISSING4
COPY C:\KEYSTROK\DATA\HOLD\KSIL*.DAT C:\KEYSTROK\DATA\KSIL*.DAT
GOTO END
:EXTRACT
COPY C:\KEYSTROK\DATA\HOLD\EXTRACT.DAT C:\KEYSTROK\DATA\EXTRACT.DAT
GOTO END
:NOFILE
PAUSE NO FILE
:END

```

# Troubleshooting

## ***I don't see new sales information – it appears that transactions are not being merged:***

Use reports such as Sales Transactions (Invoices – Summary) to verify which transactions have been polled/merged.

Verify the sized of the file **KSTRN001.MRG** in the each store's subdirectory. This file should be greater than 2,048 bytes (a file that is 2,048 bytes is an empty file). Each record contains 128 bytes, so a file with 100 records would be 3,328 bytes. This should give you a good idea of the number of records in each file.

Check the time and date of the file **KSTRN001.MRG** to determine when it was created. Was it created during the last polling session?

Make sure the polling was run at the correct time of day. Most polling scripts use the **END=M** switch, which defaults to the nearest Midnight. If polling is run before 12:00 Noon, no transactions for that day will be polled. If the **END=M** switch is NOT used, make sure the polling is run after 12:00 Midnight to get the previous days information.

## ***I received an error message during polling and don't believe I got all the data.***

Run reports to determine exactly which transactions where polled. Compare reports run from the remote location to reports run from the subdirectory for that store. By reviewing the totals at the bottom of the report, you can compare the number of Invoices and total dollars for a given time period. Reports should be run in both the Master database (**DATA\**), as well as the store's subdirectory (**STORE2\**).

## ***It appears I missed yesterday's transactions. Will the polling get them next time?***

More often than not, the polling process will get the previous transactions that were missed. To verify this. View/Edit the file **DATA\EXTRACT.DAT** at the remote location. Verify that the **START=** line has the correct date. For example, if you are missing transactions from 4/10/02 and today is 4/11/02, the line in **EXTRACT.DAT** should read **START=04/10/02**. If it does not, edit the file.

---

**NOTE:** Because most polling scripts use the **END=M** switch, this procedure would not be used to get 2/28/02 information on 4/11/02 if other transactions have since been polled. A manual Extract/StoreMerge would be required.

---

## ***I get a "Duplicate Serial Numbers Were Found" message using Quantity (QOH) Lookup***

**( F5):**

Make sure the file **KEYSTROK.LIC** from the License Disk (or Disk#1) is copied to the appropriate subdirectory at the Master location before polling takes place.

Check the Store Number under Company – Name/Address in Configuration Manager at the remote location and also in the subdirectory for that store.

You may have to delete the files **KSIL\*.DAT** from all locations (**KEYSTROK\**, **STORE1\**, **STORE2\** directories at the Master and in the **DATA\** directory at each remote location). The polling procedure will recreate these files.

***My Customer Next Number at Store #1 is being reset to what is in Store #2***

Verify that the Customer Next Number is set correctly at the remote location.

Make sure **COPYNUM.EXE** is being used.

**CHKNUM.EXE** can be added to existing polling to help avoid and identify the problem. The file **CHKNUM.LOG** is created; review the file **DOC\CHKNUM.DOC** for an example of its use.

***New Customers are disappearing***

Make sure that either the Last Purchase or Last Payment date on the Customer Default screen is not set to “0” (zero), which will put the date of “/ /” in that field. The **EXTRACT.EXE** program checks this field to determine if a Customer’s information is included in the polling. These fields normally get updated automatically, but would not if a Customer record is added but no transaction is created.

***I have duplicate transactions when I run reports and edit Invoices.***

The utility **REMDUPES.EXE** can be used to remove duplicate transactions. This utility should be used as a last resort. Finding the cause of the duplicates is recommended.

From a DOS Prompt, run Remdupes with the following optional switches:

**/D <path>** – Data Directory where duplicates exist.

**/NOINV** – Don’t update Inventory information. (If the /NOINV switch was used when STOREMRG.EXE was run.)

**Example:**

**C:\KEYSTROK>REMDUPES.EXE /DSTORE2** – This will remove duplicates from the **Store2** subdirectory under **KEYSTROK\** and reverse the quantity information on the Inventory records.

or

**C:\KEYSTROK>REMDUPES.EXE /DDATA /NOINV** – This will remove duplicates from the data subdirectory under **KEYSTROK\** and WILL NOT reverse the quantity information on the Inventory records.

Using the /NOINV switch should also be used in QOH are correct, possibly due to an adjustment to Inventory (Variance, manually editing QOH, etc.).

# Variations and Restrictions

## Variations

Another recommended option for consolidating sales transactions is to transfer sales and Accounts Receivable data for each store into a common General Ledger (i.e., BusinessWorks or CYMA PASIII) using the GL-Link module. This approach will provide financial figures for all stores combined, as well as for individual stores. Using this technique, the **Keystroke** system is operated independently at each store or location. Periodically (daily, weekly, or monthly), a journal entry file is generated at each store by using the GL-Link module. The journal entry file from each store is then transferred and imported to a single General Ledger. Depending on how the chart of accounts is set up, journal entries can either be combined into common accounts or imported into separate sub-accounts for each store.

## Restrictions



Managing data for multiple locations does involve some restrictions. Most of these restrictions are due to simple logistics, such as the difficulty of independently and simultaneously making changes to two copies of the same data, and then consolidating those changes. Therefore, certain actions (such as editing Inventory records) must only be done at the Master location. If you are planning on using any of the MultiStore tools, all Inventory additions/changes should be made in the Master store and then transferred to the individual stores.

If you plan on consolidating sales and/or Customer information from different stores, you should make sure each store uses a different set of transaction numbers and Customer numbers. In order to set these numbers, you must change the number counters in each individual store's data files. The Invoice numbers are set using the Number/Date function in the Transaction Manager; Customer Numbers are set using the Next Number function in the Database Manager. For example, Store 1's Invoices should start on Invoice number 1,000,000 and any new Customers should start on Customer number 10,000; Store 2's Invoices should start on Invoice number 2,000,000 and any new Customers should start on Customer number 20,000, etc.

Editing or modifying already polled Orders (Sales and Purchases) can cause errors on Inventory quantities (Allocated, On Order, etc.). It is therefore recommended to add a new Order/Invoice when differences occur. For example, a Purchase Order is placed for 10 items, but you actually receive 11. Instead of editing the original order, a new Purchase Invoice should be placed for the additional item.

Deletion of transactions is not supported through the MultiStore utilities. Therefore, if a transaction is deleted at a remote location, it will need to be deleted in that store's subdirectory at the Master location, and from the Master database as well. It is recommended that you use the "Void" method of deleting transactions, which actually creates a reversing transaction with the current date/time.

Because MultiStore installations often span different counties and states, Tax Tables and Tax Districts should be well thought out, and different Tables should be used at each location. Setting a Customer's Tax Table to an asterisk (\*) will cause the program to use the default Tax Table for the store in which the purchase is being made.

ROA payments may not have any Invoices applied to them (the entire amount must be a Credit Memo). The best way to assure this is to lock everyone out of the Accounts Receivable Module at the remote location and force that location to enter ROA payments using the Credit Memo function ( ) in the Sales Manager.

# Using the Keystroke General Ledger Link with the MultiStore Module

If you are also using **Keystroke's** General Ledger Link (GL-Link) module, you will need to decide how you will set up and use it in conjunction with the MultiStore module. The decisions you make will depend on how you have set up your accounting package, and what accounting methods your accountant has chosen for your business. You will need to discuss the following options with your CPA/Accountant prior to setting up the GL-Link accounts and MultiStore.

## Separate Accounts

Most companies will account for each store separately within their own General Ledger. By using sub-accounts for each store, you can separate each store's sales, inventory, and cost of goods accounts. This will result in more detail when running reports.

Because most General Ledger programs can combine sub-accounts during the reporting process, starting with separate accounts for each store location is most common, and the preferred method. This allows the tools in your General Ledger program to do the combining.

**NOTE:** You should ask your CPA/Accountant for help in determining which General Ledger accounts would be used for activities done within **Keystroke POS**. Many times your accountant can print a Chart of Accounts from your accounting package and indicate the appropriate accounts.

Below is a sample listing of the Chart of Accounts for 2 store locations, using QuickBooks Pro 2002.

Name	Type	Balance
Checking Account	Bank	1,573,933.48
Store 1 Account	Bank	1,191,602.62
Store 2 Account	Bank	382,330.86
Inventory	Other Current Asset	5,563,872.53
Store 1	Other Current Asset	4,351,336.41
Store 2	Other Current Asset	1,212,536.12
Keystroke AR	Other Current Asset	16,284,540.30
Store 1 AR	Other Current Asset	16,309,890.40
Store 2 AR	Other Current Asset	-25,350.10
Other Payments	Other Current Asset	73,226.94
Store 1 Other Payments	Other Current Asset	73,226.94
Store 2 Other Payments	Other Current Asset	0.00
Keystroke Deposits	Other Current Liability	1,052,384.30
Store 1 Deposits	Other Current Liability	799,086.30
Store 2 Deposits	Other Current Liability	253,298.00
KS Purchases	Other Current Liability	3,593,192.17
Store 1 Purchases	Other Current Liability	2,396,921.07
Store 2 Purchases	Other Current Liability	1,196,271.10
KS Sales Tax	Other Current Liability	16,202,223.83
Store 1 Tax Received	Other Current Liability	16,197,953.13
Store 2 Tax Received	Other Current Liability	4,270.70
Payroll Liabilities	Other Current Liability	0.00
Opening Bal Equity	Equity	0.00
Retained Earnings	Equity	0.00
KS Finance Charges	Income	
Store 1 Finance	Income	
Store 2 Finance	Income	
KS Shipping	Income	
Store 1 Shipping	Income	
Store 2 Shipping	Income	
Sales	Income	
Store 1 Sales	Income	
Store 2 Sales	Income	
Sales Discounts	Income	
Store 1 Discount	Income	
Store 2 Discount	Income	
Cost of Goods Sold	Cost of Goods Sold	
Store 1 COGS	Cost of Goods Sold	
Store 2 COGS	Cost of Goods Sold	
Automobile Expense	Expense	

Examples of Income Accounts in your General Ledger might be:

Sales (Main Account)	Income Account
Store 1 Sales (Sub-Account)	Income Account
Store 2 Sales (Sub-Account)	Income Account

### ***Setting Up GL-Link in Keystroke POS***

General Ledger accounts will need to be set up separately in each of the store subdirectories. For example, in a standard MultiStore setup, data would be located in **DATA\**, **STORE1\**, and **STORE2\** directories. You will need to set up GL Accounts in both the **STORE1\** and **STORE2\** data sets.

Open **Keystroke** using the Store1 data (**WINSTROK.EXE /D STORE1**). Based on the above account example and using QuickBooks Pro 2002, you would enter your accounts like this:

Sales Subtotal:	Sales:Store 1 Sales
Discount:	Sales Discounts:Store 1 Discount
Shipping:	KS Shipping:Store 1 Shipping
Sales Tax:	KS Sales Tax:Store 1 Tax Received
Cash Payments:	Checking Account:Store 1 Account
Credit Payments:	Keystroke AR:Store 1 AR
Other Payments:	Other Payments:Store 1 Other Payments
Inventory:	Inventory:Store 1
Cost of Goods Sold:	Cost of Goods Sold:Store 1 COGS

---

**NOTE:** *These examples are for QuickBooks. The General Ledger accounts are separated from their sub-accounts by the use of a colon (:) when importing entries.*

---

Within the Store2 data set (**WINSTROK.EXE /D STORE2**), enter the accounts in the same manner, indicating this is Store 2 data:

Sales Subtotal:	Sales:Store 2 Sales
Discount:	Sales Discounts:Store 2 Discount
Shipping:	KS Shipping:Store 2 Shipping
Sales Tax:	KS Sales Tax:Store 2 Tax Received
Cash Payments:	Checking Account:Store 2 Account
Credit Payments:	Keystroke AR:Store 2 AR
Other Payments:	Other Payments:Store 2 Other Payments
Inventory:	Inventory:Store 2
Cost of Goods Sold:	Cost of Goods Sold:Store 2 COGS

The screen shot below shows an example of this type of setup. Note that the display box truncates the account name; however, **Keystroke POS** can save up to 50 characters in each field.

```

Posting Account Numbers
Sales Subtotal: [SALES:STORE 1 SALES ]
Discount: [SALES DISCOUNTS:STOR]
Shipping: [KS SHIPPING:STORE 1 ]
Sales Tax: [KS SALES TAX:STORE 1 ]
Cash Payments: [CHECKING ACCOUNT:STO]
Credit Payments: [KEYSTROKE AR:STORE 1 ]
Other Payments: [OTHER PAYMENTS:STORE]
Inventory: [INVENTORY:STORE 1 ]
Cost of Goods Sold: [COST OF GOODS SOLD:S]

Finance Charges: [KS FINANCE CHARGES:S]
Layaway Payments: [KEYSTROKE DEPOSITS:S]
Order Deposits: [KEYSTROKE DEPOSITS:S]
Paid Outs: [OTHER EXPENSES:STORE]
Inventory Variance: [INVENTORY VARIANCE:S]
Purchases Subtotal: [KS PURCHASES:STORE 1 ]

< Ok > < Cancel >

```

### Store Transfers

Store transfers can be posted to the inventory accounts for each store. This process will create an import file to transfer inventory values from one account to the other.

To post Store Transfers, you must have already set up the Inventory Accounts within each of the store's data sets (i.e. /D **STORE1**) within the GL-Link module. The store transfer option will only be seen when you enter the GL-Link using the **DATA\** data set, even though you will set up the accounts in each data set (by using the /D switch).

After entering the GL-Link module using the main **DATA\** data set, select the **Transfers...** option located under the Link menu. Enter the dates you want to post and press **ENTER**. This will create a file similar to the screen shot below.

Entry#	Date	Referenc	Description	Account	Amount
1	06/29/00	TRANSFER	#1 to #2	INV:STORE1	-142.13
				INV:STORE2	142.13
2	07/21/00	TRANSFER	#1 to #2	INV:STORE1	-1.96
				INV:STORE2	1.96

## **Combined Accounts**

In rare cases, you may choose to not use separate accounts for each store. This decision should be made by your CPA/Accountant and is not very common. This is done by combining all sales into a General Ledger account (one for Sales, one for combined Inventory, one for combined Cost of Goods Sold, etc.) This method does not provide for detailed reporting based on each individual store.

An example of using combined accounts would be if you operate an outdoor kiosk. You and your Accountant have decided not to document this kiosk as a separate company, and you want to consolidate the sales from the kiosk with the sales from the main store.

In this case, you would post to accounts that are set up in the Master or consolidated database (**DATA\**). Open **Keystroke** using the Master data set (**WINSTROKE.EXE /D DATA**), and enter the General Ledger accounts as follows:

<b>Sales Subtotal:</b>	<b>Sales</b>
<b>Discount:</b>	<b>Sales Discounts</b>
<b>Shipping:</b>	<b>KS Shipping</b>
<b>Sales Tax:</b>	<b>KS Sales Tax</b>
<b>Cash Payments:</b>	<b>Checking Account</b>
<b>Credit Payments:</b>	<b>Keystroke AR</b>
<b>Other Payments:</b>	<b>Other Payments</b>
<b>Inventory:</b>	<b>Inventory</b>
<b>Cost of Goods Sold:</b>	<b>Cost of Goods Sold</b>

---

***NOTE:** The examples above apply specifically to the QuickBooks program. Many accounting packages use account numbers instead of account names.*

---

You are now ready to create the Import file from the GL-Link module.

# Index

## Symbols & Special Keys

/D – Data Directory 15  
/FORCEDATE – Force Ending Date 13  
/LOWMEM – Low Memory 15  
/M – Merge File Directory  
    CopyClock Utility 32  
    CopyNum Utility 33  
    Extract Utility 13  
    StoreMerge Utility 15  
/NOAUDIT – No Audit 15  
/NOCUST – No Customer Update  
    Extract Utility 13  
    StoreMerge Utility 15  
/NODISPLAY – No Display 13  
/NOINV – No Inventory Update 15  
/NOLOCK – No Lock 13, 14  
/NONEXTNUMS – No Next Numbers 15  
/NOP – No Pause 15  
/NOTIMECLOCK – No Time Clock 15  
/NOVAR – No Variance  
    Extract Utility 13  
    StoreMerge Utility 15  
/NUMBERS – Numbers  
    Extract Utility 13  
    StoreMerge Utility 15  
/SALESONLY – Sales Information Only 15  
>FileName Parameter 33  
⌨ – QOH Lookup 8, 17, 64

## A

Accounts Receivable 5  
Add Allocated to QOH Parameter 29  
Add Layaway to QOH Parameter 29  
Add New Items  
    Inventory Update Parameters 25  
Adding New Inventory Items 28  
Adding Stores 27  
AddSubjectToPath=  
    GetMail (GETMAIL.EXE) 34  
All Items  
    Merge Same Items Parameter 30  
Attachment=  
    SendMail (SENDMAIL.EXE) 36  
AttachmentPath=  
    GetMail (GETMAIL.EXE) 34  
Avg Cost  
    Inventory Update Parameters 26

## B

Base Price  
    Inventory Update Parameters 26  
Blast Data Pump 9, 42  
Blast Professional for DOS 9, 42  
BLASTPOL.ZIP 42

## C

Change Clerk Function 27  
ChangeNums (CHGNUMS.EXE) 6, 31  
CheckNum (CHKNUM.EXE) 6, 31  
CheckScript (CHKSCRIPT.EXE) 6, 32  
CheckStamp (CHKSTAMP.EXE) 6, 7, 36–37  
CHGNUMS.EXE. *See* ChangeNums  
    (CHGNUMS.EXE)  
CHKFTP.BAT 62  
CHKNUM.EXE. *See* CheckNum (CHKNUM.EXE)  
CHKNUM.LOG 31  
CHKSCRIPT.EXE. *See* CheckScript  
    (CHKSCRIPT.EXE)  
CHKSTAMP.EXE. *See* CheckStamp  
    (CHKSTAMP.EXE)  
CHKSTAMP.LOG 36  
Clear Function 20  
CloseUp 9, 42  
Comment  
    Inventory Update Parameters 26  
Commission Code  
    Inventory Update Parameters 26  
Communications Software 8  
Confirm Cost  
    Entry Parameters 23  
Confirm Qty  
    Entry Parameters 23  
Copy Function 20, 21  
Copy Inventory Comment  
    Entry Parameters 24  
COPYCLK.EXE. *See* CopyClock (COPYCLK.EXE)  
CopyClock (COPYCLK.EXE) 6, 32  
CopyNum (COPYNUM.EXE) 6, 12, 15, 32–33  
COPYNUM.EXE. *See* CopyNum (COPYNUM.EXE)  
Credit Memo Function 66  
Current Function 18, 20, 26–27  
Customer Number 32, 66

## D

Data Configurations 9–20  
  Main Store Master 9, 10  
  Virtual Master 9, 9–10  
  Warehouse Master 9, 11–20  
DateName (DATENAME.EXE) 6, 33  
DATENAME.EXE. *See* DateName  
  (DATENAME.EXE)  
DATENAME.TXT 6, 33  
Default to Save  
  Entry Parameters 24  
Delete Function 20, 21  
Deleting Stores 27  
Deleting Transactions 66  
Don't Update Master Store QOH  
  Entry Parameters 24  
DOS ErrorLevel 31, 33, 36  
Duplicate Items 30  
“Duplicate Serial Numbers Were Found” Message  
  64–65

## E

E-Commerce 34, 36  
Edit/Print Function 20, 21  
Editing Stores 27  
Email. *See* GetMail (GETMAIL.EXE); SendMail  
  (SENMAIL.EXE). *See also* MultiStore  
  Polling via Email  
END={date} 13, 64  
ENDCLERK={Ending Clerk Number}  
  Extract Utility 14  
  StoreMerge Utility 16  
ENDREG={Ending Register Number}  
  Extract Utility 14  
  StoreMerge Utility 16  
ENDSTORE={Ending Store Number}  
  Extract Utility 14  
  StoreMerge Utility 16  
ENDTIME={time} 13  
ENDTRANS={Ending Transaction Number}  
  Extract Utility 14  
  StoreMerge Utility 16  
Enter Store First  
  Entry Parameters 23  
Enter Store Information Screen 28  
Entry Parameters 23–25  
  Confirm Cost 23  
  Confirm Qty 23  
  Copy Inventory Comment 24  
  Default to Save 24  
  Don't Update Master Store QOH 24  
  Enter Store First 23  
  Item Field 23  
  No Out of Stock Transfers 23  
  Print and Continue 24  
  Reset Avg. Cost 25  
  Update Avg. Cost 24  
  Update Last Cost 24

Error Checking Utilities. *See* Miscellaneous/Error  
  Checking Utilities  
Exclude from Inventory Update 28  
Exclude from Levels Report 28  
EXCLUDE={CV1WQO5RYyA0T9s}  
  Extract Utility 13  
  StoreMerge Utility 15, 16  
Exit Menu 20  
Export Function 31  
EXT1.BAT 50, 55, 57  
EXT1.INI 51, 55  
EXT2.BAT 55, 57  
EXT2.INI 55  
Extract (EXTRACT.EXE) 5, 6, 7, 12–  
  14, 14, 32, 64, 65  
  Optional Switches  
    /FORCEDATE – Force Ending Date 13  
    /M – Merge File Directory 13  
    /NOCUST – No Customer Update 13  
    /NODISPLAY – No Display 13  
    /NOLOCK – No Lock 13  
    /NOVAR – No Variance 13  
    /NUMBERS – Numbers 13  
  END={date} 13  
  ENDCLERK={Ending Clerk Number} 14  
  ENDREG={Ending Register Number} 14  
  ENDSTORE={Ending Store Number} 14  
  ENDTIME={time} 13  
  ENDTRANS={Ending Transaction Number} 14  
  EXCLUDE={CV1WQO5RYyA0T9s} 13  
  EXTRACT={CV1WQO5RYyA0T9s} 13  
  INCLUDE={CV1WQO5RYyA0T9s} 13  
  START={date} 13  
  STARTCLERK={Starting Clerk Number} 14  
  STARTREG={Starting Register Number} 14  
  STARTSTORE={Starting Store Number} 14  
  STARTTIME={time} 13  
  STARTTRANS={Starting Transaction Number} 14  
EXTRACT.DAT 13, 14, 38, 64  
EXTRACT.DOC 12  
EXTRACT.EXE. *See* Extract (EXTRACT.EXE)  
EXTRACT={CV1WQO5RYyA0T9s}  
  Extract Utility 13  
  StoreMerge Utility 16

## F

FileMask 33  
Filter Parameter  
  Inventory Function 29  
  Transfers Function 29  
FindFile (FINDFILE.EXE) 6, 33  
FINDFILE.EXE. *See* FindFile (FINDFILE.EXE)  
Form Files 21  
Form Parameter 29  
FTP. *See* MultiStore Polling via FTP

## G

General Ledger 66  
General Ledger Link 67–70. *See also* GL-Link  
GETINV1.BAT 54, 56, 57  
GETINV1.INI 55, 56  
GETINV2.BAT 56, 57  
GETINV2.INI 56  
GetMail (GETMAIL.EXE) 6, 34, 50  
GETMAIL.INI 34  
GETSALE1.INI 53, 56  
GETSALE2.INI 56  
GL-Link 5, 8. *See also* General Ledger Link  
Group By Parameter 29

## I

Import Line Items Function 20, 30  
Importer Module 31  
In Stock Only Parameter 29  
Include Service Items Parameter 29  
INCLUDE={CV1WQO5RYyA0T9s}  
    Extract Utility 13  
    StoreMerge Utility 15, 16  
Inventory Fields  
    Inventory Update Parameters 26  
Inventory Function 20, 29  
    Add Allocated to QOH Parameter 29  
    Add Layaway to QOH Parameter 29  
    Filter Parameter 29  
    Group By Parameter 29  
    In Stock Only Parameter 29  
    Include Service Items Parameter 29  
    Print Class Parameter 29  
    Print To Parameter 29  
    Sort By Parameter 29  
    Use InvLink Files Parameter 29  
Inventory Link (INVLINK.EXE) 5, 6, 8, 17–18, 29  
Inventory Transfers 18–19  
Inventory Update Parameters 25–26  
    Add New Items 25  
    Inventory Fields 26  
    Matrix Items 26  
    Remove Deleted Items 25  
    Update Existing Items 26  
InvLink Files. *See* InventoryLink (INVLINK.EXE)  
INVLINK.DAT 17  
INVLINK.EXE. *See* Inventory Link (INVLINK.EXE)  
Invoice Number 32, 66  
ISDN 9  
Item Field  
    Entry Parameters 23

## J

Journal Entries 8, 66

## K

KEYSTROK.LIC 17, 64  
Keystroke Polling (KSSPOL.BAT) 35  
KS Scheduler (KSSCHDLR.EXE) 7, 35  
KSCLK.DAT 6, 32  
KSCLK.TMP 6, 32  
KSCST.DAT 6, 32  
KSIL#.DAT 6, 8, 17. *See also* InventoryLink  
    (INVLINK.EXE)  
KSIL\*.DAT 29, 64. *See also* KSIL#.DAT  
KSPOLL.BAT 35  
KSSCHDLR.EXE. *See* KS Scheduler  
    (KSSCHDLR.EXE)  
KSSTORES.DAT 27  
KSSTR\*.DAT 30  
KSTRN001.MRG 6, 7, 12, 13, 14, 15, 32, 64

## L

Last Cost  
    Inventory Update Parameters 26  
List Price  
    Inventory Update Parameters 26

## M

Main Store Master 9, 10  
Managing Data Files 11–19  
    Inventory Data 17–18  
    Store Data 18–19  
    Transaction Data 12–16  
Matrix Items  
    Inventory Update Parameters 26–38  
Matrix Manager 31  
Menus  
    Exit Menu 20  
    Reports Menu 20, 29  
    Special Menu 20, 30  
    Store Menu 20, 26–28  
    Transfer Menu 20, 20–26  
Merge Same Items 30  
MERGE1.ZIP 55, 56  
MERGE2.ZIP 55, 56  
MERGEAW4.ZIP 42  
MERGECU4.ZIP 42  
Minimum Qty  
    Inventory Update Parameters 26  
Miscellaneous/Error Checking Utilities 31–37  
Modems 9  
MRGDPPOL.ZIP 42  
MRGEMAIL.ZIP 42, 55  
MRGFTP.ZIP 42  
MRGPCA92.ZIP 42  
MRGPCAW8.ZIP 42  
MRGSALES.BAT 51, 56, 57  
MultiStore Polling via Email 50–57  
MultiStore Polling via FTP 58–63

## N

New Items Only 30  
Next Number 31  
Next Number Function 32, 66  
No Out of Stock Transfers  
    Entry Parameters 23  
NoPause=  
    GetMail (GETMAIL.EXE) 34  
NoteText=  
    GetMail (GETMAIL.EXE) 34  
    SendMail (SENDMAIL.EXE) 36  
Number/Date Function 20, 22, 32

## O

On Sale Flag  
    Inventory Update Parameters 26  
Order Qty  
    Inventory Update Parameters 26  
Overview 7–8

## P

Pack Function 20, 30  
Parameters Function 20, 22–26  
    Entry Parameters 23–25  
    Inventory Update Parameters 25–26  
Password=  
    GetMail (GETMAIL.EXE) 34  
    SendMail (SENDMAIL.EXE) 36  
pcAnywhere 9, 42  
    Sample Scripts 48–63  
Phone Lines 9  
Polling Procedures 42  
Polling Script 5, 7, 8, 9, 12, 14, 42, 64  
    Samples 43–44, 45–63  
Price Code  
    Inventory Update Parameters 26  
Price Level 1  
    Inventory Update Parameters 26  
Price Level 2  
    Inventory Update Parameters 26  
Price Level 3  
    Inventory Update Parameters 26  
Print and Continue  
    Entry Parameters 24  
Print Button 21, 24  
Print Class Parameter 29  
Print To Parameter  
    Inventory Function 29  
    Transfers Function 29  
Print/Save Function 20, 22  
Printing  
    Store Transfers 21  
Production Module 31

## Q

QOH Lookup –  8, 17, 64  
QuickBooks Pro 2002 68

## R

Recall Function 20, 22  
RemDupes (REMDUPES.EXE) 7, 35, 65  
REMDUPES.EXE. *See* RemDupes (REMDUPES.EXE)  
Remove Deleted Items  
    Inventory Update Parameters 25  
Reports Menu 20, 29  
    Inventory Function 20, 29  
    Transfers Function 20, 29  
Requirements 8–9  
Reset Avg. Cost  
    Entry Parameters 25  
Restrictions. *See* Variations & Restrictions  
ROA Payments 66

## S

Sale Price  
    Inventory Update Parameters 26  
Sample Polling Procedures 42–70  
SaveTo=  
    GetMail (GETMAIL.EXE) 34  
Script Editor Module 31  
SearchText 33  
SEND2.ZIP 59, 62  
SENDINV.BAT 56, 57  
SENDINV1.BAT 53, 56  
SENDINV1.INI 54, 56  
SENDINV2.BAT 56  
SENDINV2.INI 56  
SendMail (SENDMAIL.EXE) 7, 36, 50  
SENDMAIL.INI 36  
SendTo=  
    SendMail (SENDMAIL.EXE) 36  
Setting up MultiStore 38–41  
Setup Function 20, 27–28  
    Enter Store Information Screen 28  
    Exclude from Inventory Update 28  
    Exclude from Levels Report 28  
Sort By Parameter 29  
Special Menu 20, 30  
    Import Line Items Function 20, 30  
    Pack Function 20, 30  
Stamp (STAMP.EXE) 6, 7, 36–37  
STAMP.EXE. *See* Stamp (STAMP.EXE)

## Stand-Alone Utilities

- ChangeNums (CHGNUMS.EXE) 6, 31
- CheckNum (CHKNUM.EXE) 6, 31
- CheckScript(CHKSCRPT.EXE) 6, 32
- CheckStamp (CHKSTAMP.EXE) 6, 36–37
- CopyClock (COPYCLK.EXE) 6, 32
- CopyNum (COPYNUM.EXE) 6, 32–33
- DateName (DATENAME.EXE) 6, 33
- Extract (EXTRACT.EXE) 6, 12–14
- FindFile (FINDFILE.EXE) 6, 33
- GetMail (GETMAIL.EXE) 6, 34
- Inventory Link (INVLINK.EXE) 6, 17–18
- Keystroke Polling (KSPOL.BAT) 35
- KS Scheduler (KSSCHDLR.EXE) 7, 35
- RemDupes (REMDUPES.EXE) 7, 35, 65
- SendMail (SENDMAIL.EXE) 7, 36
- Stamp (STAMP.EXE) 7, 36–37
- Store Update (STRUPDTE.EXE) 7, 37
- StoreMerge (STOREMRG.EXE) 7, 14–16
- START={date} 13, 64
- STARTCLERK={Starting Clerk Number}
  - Extract Utility 14
  - StoreMerge Utility 16
- STARTREG={Starting Register Number}
  - Extract Utility 14
  - StoreMerge Utility 16
- STARTSTORE={Starting Store Number}
  - Extract Utility 14
  - StoreMerge Utility 16
- STARTTIME={time} 13
- STARTTRANS={Starting Transaction Number}
  - Extract Utility 14
  - StoreMerge Utility 16
- Stock Numbers 19
- Store Menu 20, 26–28
  - Current Function 18, 20, 26–27
  - Setup Function 20, 27–28
  - Transfer To Function 20, 27
  - Update Inventory Function 20, 28
- Store Number 28
- Store Setup List. *See* Setup Function
- StoreLink 5, 7, 8, 11, 18–19, 20–30, 37
  - Clear Function 20
  - Copy Function 20, 21
  - Current Function 20, 26–27
  - Delete Function 20, 21
  - Edit/Print Function 20, 21
  - Import Line Items Function 20, 30
  - Inventory Function 20, 29
  - Menus 20–30
    - Exit Menu 20
    - Reports Menu 20, 29
    - Special Menu 20, 30
    - Store Menu 20, 26–28
    - Transfer Menu 20, 20–26

## StoreLink (cont.)

- Number/Date Function 20, 22
- Pack Function 20, 30
- Parameters Function 20, 22–26
- Print/Save Function 20, 22Recall Function 20, 22
- Setup Function 20, 27–28
- Transfer To Function 20, 27
- Transfers Function 20, 29
- Update Inventory Function 20, 28
- StoreMerge (STOREMRG.EXE) 5, 6, 7, 12, 14–16, 64
  - Optional Switches
    - /D – Data Directory 15
    - /LOWMEM – Low Memory 15
    - /M – Merge File Directory 15
    - /NOAUDIT – No Audit 15
    - /NOCUST – No Customer Update 15
    - /NOINV – No Inventory Update 15
    - /NONEXTNUMS – No Next Numbers 15
    - /NOP – No Pause 15
    - /NOTIMECLOCK – No Time Clock 15
    - /NOVAR – No Variance 15
    - /NUMBERS – Numbers 15
    - /SALESONLY – Sales Information Only 15
  - ENDCLERK={Ending Clerk Number} 16
  - ENDREG={Ending Register Number} 16
  - ENDSTORE={Ending Store Number} 16
  - ENDTRANS={Ending Transaction Number} 16
  - EXCLUDE={CV1WQO5RYyA0T9s} 16
  - EXTRACT={CV1WQO5RYyA0T9s} 16
  - INCLUDE={CV1WQO5RYyA0T9s} 16
  - STARTCLERK={Starting Clerk Number} 16
  - STARTREG={Starting Register Number} 16
  - STARTSTORE={Starting Store Number} 16
  - STARTTRANS={Starting Transaction Number} 16
- STOREMRG.DAT 16
- STOREMRG.EXE. *See* StoreMerge (STOREMRG.EXE)
- StoreUpdate (STRUPDTE.EXE) 7, 37
- .STR Files 21
- STRUPDTE.EXE. *See* StoreUpdate (STRUPDTE.EXE)
- Subject=
  - GetMail (GETMAIL.EXE) 34
  - SendMail (SENDMAIL.EXE) 36

## T

- Tax Code
  - Inventory Update Parameters 26
- Tax Districts 66
- Tax Tables 66
- .TMP Files 30
- Transaction Lock date 13
- Transfer Menu 20, 20–26
  - Clear Function 20
  - Copy Function 20, 21
  - Delete Function 20, 21
  - Edit/Print Function 20, 21
  - Number/Date Function 20, 22
  - Parameters Function 20, 22–26
  - Print/Save Function 20, 22
  - Recall Function 20, 22
- Transfer To Function 20, 27
- Transfers Function 29
  - Filter Parameter 29
  - Form Parameter 29
  - Print To Parameter 29
- Troubleshooting 64–65

## U

- Update Avg. Cost
  - Entry Parameters 24
- Update Existing Items
  - Inventory Update Parameters 26
- Update Inventory Function
  - 5, 7, 12, 20, 22, 25, 26, 27, 28, 37, 40
- Update Last Cost
  - Entry Parameters 24
- Updating Existing Inventory Items 28
- Use InvLink Files Parameter 29
- UserName=
  - GetMail (GETMAIL.EXE) 34
  - SendMail (SENDMAIL.EXE) 36

## V

- Variance Function 12
- Variations & Restrictions 66–70
- Virtual Master 9, 9–10

## W

- WAN (Wide Area Network) 7, 8
- Warehouse Master 9, 11–20
- Windows Scheduler 35