Enterprise Extract

Infinity Software Solutions, Inc
1414 19th Street
Palm Harbor, FL 34683
727-785-8563
infinity_software@verizon.net

Create Enterprise Extract Program

While testing the COLA manager discovered that the path generated for the SAMOHF file was incorrect.

He was right. There was a bug in the extract program which we corrected and delivered on the same day.

Files affected by the enhancement

Source files altered:	SYEEXX.CBL	Extract program
Files to the customer:	sy\SYEEXX.GNT	Extract program
	Readme.doc	This file

Date: 5/2/2014
ISS#: ISS851 (continued)

Create Enterprise Purge Program

The last version of the purge process worked well, but the client requested that we move the original file to the directory identified by the ARCHIVE directory. Previously we had added code to use this same environment variable to save the archived data files.

The program now copies the original file to a subdirectory named after the file's original data directory within the ARCHIVE directory. So the customer file gets backed up in the *ARCHIVE*\ARDATA directory. If the data directory doesn't exist the program will create it.

Files affected by the enhancement

Source files altered:	SYPRGU.CBL	Extract program
Files to the customer:	sy\SYPRGU.GNT	Extract program
	Readme.doc	This file

Date: 5/2/2014
ISS#: ISS851 (continued)

Create Enterprise Purge Program

Some of the clients' customers want to be able to view their ticket history after the purge. The way the client asked us to create the initial version of the purge process wouldn't allow this. That version merely read through the history file and wrote the records to be retained to a new file. Both files were then closed. The original file was then renamed to a backup file name (in place) and the new file was renamed to the production name.

We modified the purge process to split the file being archived rather than merely copying the records that they want to retain from the files. This creates an archive file of the older data, a new file containing the records to be retained going forward and leaves the original file as-is. The original file which contains all of the data is renamed in place as it was in the initial version of the purge process. The new data file is renamed to the production filename (as in the initial version of the program) and also stays in the production directory.

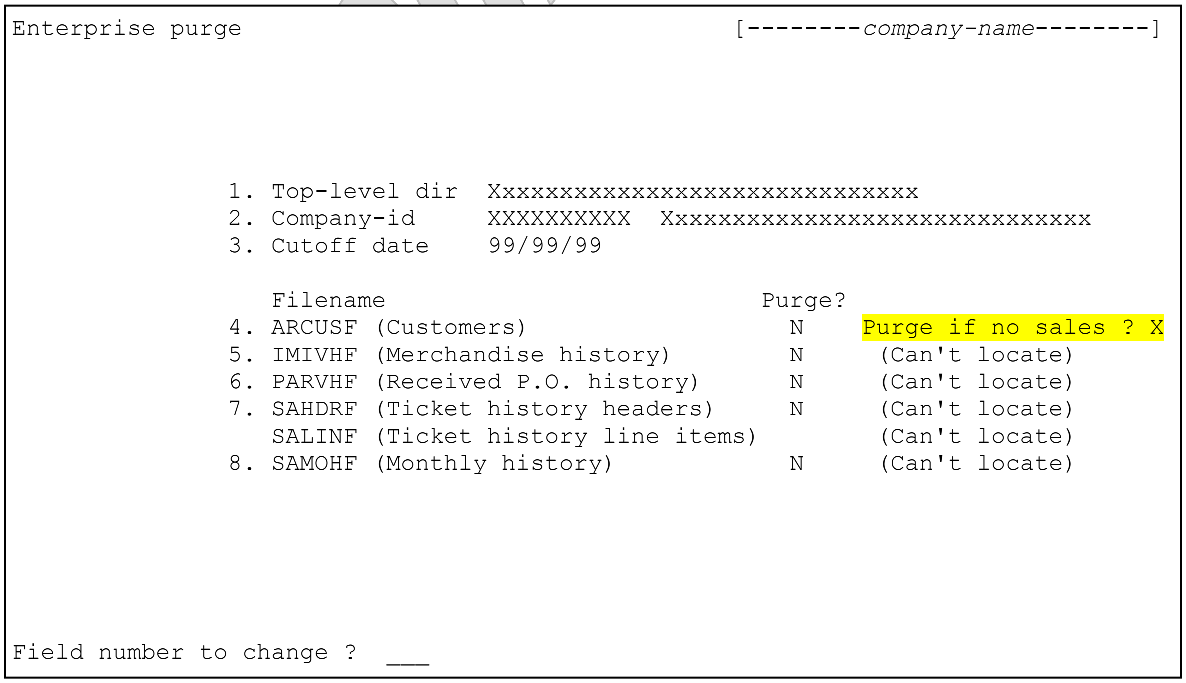
The archive files are created in a directory pointed to by the environment variable ARCHIVE. When the purge parameter entry program first starts it checks that the environment variable exists. If it does not, the program exits with an error message. The program concatenates the value of the ARCHIVE environment variable and the company-id parameter and checks that the directory structure exists. If it doesn't the program offers to create the directory structure. If the operator declines the program does not proceed. The archived files are not renamed once they are created. They are created with the same name as the original file they came from. The archived files retain the structure of their original file. So they could actually be copied to their original production directories and viewed.

File Utilities>System>Enterprise purge

As well as the changes to the basic functionality of the purge process (as described in the opening paragraphs) the client asked us to add special code when purging the customer file. Customers that have never made a purchase have zeros in their last-purchase-date field. All of these customers would naturally be archived regardless of the cut-off date used in the purge. The client asked that we add a way to control this behavior.

We added a sub parameter to the Purge customer parameter in the purge parameter program. If the client answers "Y" to purge the customer file the program asks "Purge if no sales ?". If the client answers "N" the program does not purge customers with no sales (their last-sales-date field is zeros).

The parameter screen now resembles:



```
Enterprise purge                                  [--------company-name--------]




               1. Top-level dir  Xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
               2. Company-id     XXXXXXXXXX  Xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
               3. Cutoff date    99/99/99

                  Filename                          Purge?
               4. ARCUSF (Customers)                  N    Purge if no sales ? X
               5. IMIVHF (Merchandise history)        N     (Can't locate)
               6. PARVHF (Received P.O. history)      N     (Can't locate)
               7. SAHDRF (Ticket history headers)     N     (Can't locate)
                  SALINF (Ticket history line items)        (Can't locate)
               8. SAMOHF (Monthly history)            N     (Can't locate)





Field number to change ?  ___
```

Sales History>View>Ticket history

We modified the view ticket history program to allow the operator to switch from using the current ticket history file to the archive file. The program offers four different views to the data – Ticket, Customer/Date, Customer/Item and Item/Ticket. We added to the first field of each view the ability to press the F2 key to toggle between displaying current information of archived information.

Sales History Ticket Lookup Program

One of the view's first fields is the ticket number. That field allows pressing the F9 data lookup key. When viewing archived data, pressing the F9 key would display the tickets from the current ticket history file. Obviously that wouldn't fly so we modified the ticket history lookup program to display the tickets from the archive file based on values passed to it from the calling program.

Sales History>View>Monthly history

We modified the view monthly history program to allow the operator to switch from using the current ticket history file to the archive file. The program offers 17 different types of history to view. The operator first selects the type of history to be viewed. To the resulting parameters screen (which depends on the type of history being viewed) we added to the first field of each screen the ability to press the F4 key to toggle between displaying current information of archived information.

The program the operator enters the parameters into is not the same program that displays the monthly history. The parameter entry program calls the monthly history display program passing to it the parameters needed to retrieve the desired monthly history record. So if the operator selects to display archive data, but the monthly history display program has saved its parameters to display the current year's information, there is a strong possibility that the display program won't find any monthly history record for the current year in the archived data. That will result in a "No monthly history on file" message. That is expected behavior. Once on the screen that displays the monthly history the operator needs to press the F2 key (for Change view") and change the Year ID field to reflect the correct year to be desired for each desired field.

A side effect of this mod is that the modification can be accessed from many areas of CounterPoint. The monthly history display program is called from many areas within CounterPoint other than View Monthly History. For example, when viewing an item in the item maintenance program the operator has the ability to press the F8 key to view the item's monthly history. This action calls the modified monthly history display program which now allows viewing archived data.

Customer History Ticket Lookup Program

One of the viewable types of monthly history is Customer. The parameter screen then prompts for the customer number to be viewed. That field allows pressing the F9 data lookup key. When viewing archived data, pressing the F9 key would display the customers from the current customer file. Obviously that may be a different set of customers than is actually in the archived monthly history file. So we modified the customer lookup program to have the ability to display the records from either the current customer file or the archived customer file as appropriate.

Files affected by the enhancement

Source files altered:	ARCUSL.CBL	Customer lookup program
	SAMOHI.CBL	Monthly history parameter entry program
	SAMOHV.CBL	Monthly history display program
	SAMOHV.WS	Monthly history display linkage copyfile
	SATIXL.CBL	Sales History ticket lookup program
	SATKHI.CBL	View ticket history program
	SYPRGO.CBL	Extract program parameter entry program
	SYPRGU.CBL	Extract program
	SYPRGU.WS	Extract program linkage copyfile

Files to the customer:	ar\ARCUSL.GNT	Customer lookup program
	sa\SAMOHI.GNT	Monthly history parameter entry program
	sa\SAMOHV.GNT	Monthly history display program
	sa\SATIXL.GNT	Sales History ticket lookup program
	sa\SATKHI.GNT	View ticket history program
	sy\SYPRGO.GNT	Extract program parameter entry program
	sy\SYPRGU.GNT	Extract program
	Install.bat	Installation script
	Readme.doc	This file

Date: 3/27/2014
ISS#: ISS849 (continued)
ISS#: ISS851 (continued)

Create Enterprise Extract Program

The client started testing the extract and found a problem. They use non-default paths for much of their CP data. The extract program didn’t include the company-id in the path and so the extract program couldn’t find the data files to be extracted.

We corrected that and made a couple of other changes as well. The client requested that we changed the environment variable it uses from EXTRACTS to COMPEXTR since EXTRACTS is being used for another purpose. We also changed the programs as required to add a note to the log file to indicate how the extract process was being run (from the command line, from the menus or using the COMPEXTR.TXT file).

Create Enterprise Purge Program

Since these process used the extract programs as its basis it suffered from the same data file path building bug as the extract programs.

So we corrected them in the same manner.

Files affected by the enhancement

Source files altered:	SYEEXO.CBL	Extract program parameter entry program
	SYEEXX.CBL	Extract program
	SYEEXX.WS	Extract program linkage copyfile
	SYPRGO.CBL	Extract program parameter entry program
	SYPRGU.CBL	Extract program
Files to the customer:	sy\SYEEXO.GNT	Extract program parameter entry program
	sy\SYEEXX.GNT	Extract program
	sy\SYPRGO.GNT	Extract program parameter entry program
	sy\SYPRGU.GNT	Extract program
	Readme.doc	This file

Date: 3/26/2014
ISS#: ISS851

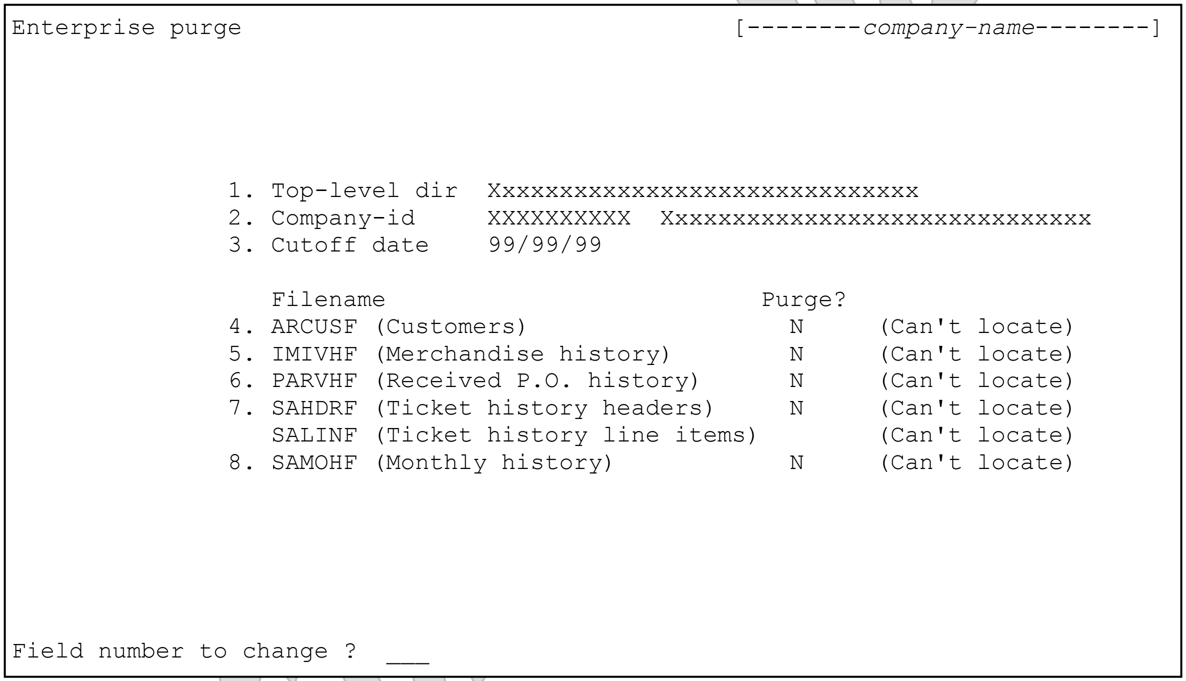
Create Enterprise Purge Program

The client is going to be extracting information from their COLO CounterPoint system nightly to create reports using an extract program we wrote recently. They will probably also continue extracting information for reports via ODBC. Their general process has slowed because their files have grown quite large over the years. They requested that we write a purge utility that will purge inactive information from the largest of CounterPoint's files.

File Utilities>System>Enterprise purge

We wrote the new utility program based off the extract programs we recently wrote for the same client. This will allow a manager to run the purge process for installations and companies other than the one he is currently in. The extract process allows running in interactive mode (from the menus) or from the command line. This utility only allows interactive mode.

The parameter screen resembles:



```
Enterprise purge                                  [--------company-name--------]




               1. Top-level dir  Xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
               2. Company-id     XXXXXXXXXX  Xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
               3. Cutoff date    99/99/99

                  Filename                          Purge?
               4. ARCUSF (Customers)                  N     (Can't locate)
               5. IMIVHF (Merchandise history)        N     (Can't locate)
               6. PARVHF (Received P.O. history)      N     (Can't locate)
               7. SAHDRF (Ticket history headers)     N     (Can't locate)
                  SALINF (Ticket history line items)        (Can't locate)
               8. SAMOHF (Monthly history)            N     (Can't locate)





Field number to change ?  ___
```

The first two parameters can be left blank.

1. Top-level dir — This parameter provides the path to the company's top-level directory. This is required when the company-id to be extracted is not part of the CounterPoint installation that is running the purge. Combining the value of this parameter with the *company-id* parameter (like this *top-level-dir/company-id*) should result in a valid pathspec for a directory that holds the SYCMPF.DAT file for the company being processed. This parameter is option. If it is omitted the program assumes that the company to be extracted is part of the same installation of CounterPoint that is running the purge program.

2. Company-id — This parameter specifies the company-id to be processed. If this parameter is missing the program assumes the company the operator is currently logged into. There should be a data directory with the same name as the company-id parameter. If there is only a single company defined for that installation of CounterPoint this directory might be named DATAFILS and that's the value that would be used.

3. Cutoff date — This parameter carries the date to be used to discern data to be retained. If a record is dated on or after this date it will be retained after the purge process. Records dated prior to is date will be purged.

The remaining parameters allow the operator to indicate which files should be purged. Once the first two parameters have been entered the program will attempt to locate and open each file the program can process. If the file cannot be located the literal ("Can't locate") will be displayed next to its parameter. If the file cannot be opened the literal ("Can't open") will be displayed next to its parameter. For each file that doesn't have such an error message next to it the program will allow entering a Y/N value to indicate whether to purge the file or not.

Overview

Processing

For each CounterPoint file processed, the program opens a new version of the file with the exact same attributes. It reads through each file evaluating a date field on each record. If the date falls before the cutoff date the program skips the record and reads the next. If the date field is greater than or equal to the cutoff date the program writes the record to the new file. Once all of the records have been evaluated the program closes both files and renames the original file to a backup name (discussed later). It then renames the new file to the original file's name. Thus the backup file will contain all of the file's original file's data. The new file will only contain records dated on or after the cutoff-date.

Backups

Once the program has completed it renames the original file's extension to a 3 digit number starting at "001" and incrementing with each run. This allows for up to 999 backups. That seems to be too many backups. So we implemented a rolling strategy. We allow the program to create the backup files numbered from 1 to 14. When the program finds that the first available file extension is "015" the program deletes the first 7 backup files and assigns the current backup the extension "001". Subsequent backups are named "002" through "007". While the program is testing for the next available extension to use it executes special code when checking for the extension "008". If the backup file exists and is dated older than "007", the program deletes the backup files numbered "008" through "014". The program then assigns the current backup file the extension "008". This methodology allows the client to retain at least 7, but not more than 14 backups of each file – which might still be a bit of an over-kill. Note that this process is executed for each file separately. It is possible for different extract files to be backed up using different file extension values.

Reporting

The program does not create a report. However it does create a log file containing pertinent messages such as the date and time it started and stopped, the number of record retained in each file and any errors encountered. This log file is named PURGE.LOG and is located in the top-level directory. Each time the program runs it appends to this file. The purge program also logs the name of each of the backup files.

The program purge information from the following files:

Filename	Description	Notes
ARCUSF	Customer File	Last activity date is evaluated
IMIVHF	Merchandise History File	See notes below
PARVHF	Receive PO History File	See notes below
SAHDRF	Ticket Header History Header	Date posted is evaluated
SALINF	Ticket Line History File	Records associated with purged headers are purged
SAMOHF	Monthly History File	See notes below

All records are evaluated. History files only have the records that match the date parameter posted. Master files (such as the item file and customer file) do not have relevant dates and so they are extracted in their entirety. If the extract is run in incremental mode, all changed records are extracted regardless of the date range parameter.

Merchandise History File (IMIVHF)

This file contains two different records types. The first type is audit records. These records have the type "A". They provide information at the most granular level (item/location/date/transaction-type) so there are many of them. The second are merchandise ("M") records. These combine the activity for each item/location/date into a single record and so there are far fewer of them. We purge both types of records that match the date parameter.

Receive PO History File (PARVHF)

This file contains the header and line item records. The program purge the records whose receipt date matches the date parameter. The client requested that we separate the headers and line items into separate files. So the program puts header records in the file RCVHDR.CSV and line items in RCVLIN.CSV.

Monthly History File (SAMOHF)

The monthly history file contains many different records types. Each type accumulates totals for a "calendar" month. All are evaluated. Since the information isn't separated by date we purge entire records based on the record's ending date. If the record's ending date is less than the cutoff date the record is purged.

CounterPoint Definition File

We added a new menu selection (*File Utilities>System>Enterprise purge*) to the CounterPoint menu definition file from which the client will run the new purge program.

Included with this proposal

- Create the new purge parameter program as described above.
- Create the new purge program as described above.
- Add the menu selection to both versions of the CounterPoint graphical menus.
- A simple installation procedure.
- Documentation of the enhancement and the files affected.
- A 30-day guarantee that the software operates as designed in this proposal.

Files affected by the enhancement

Source files altered:	CHKPTH.PL	Directory checking/creation routines
	CHKPTH.WS	
	SYPRGO.CBL	Extract program parameter entry program
	SYPRGU.CBL	Extract program
	SYPRGU.WS	Extract program linkage copyfile
	SYMENU.DAT	Menu definition file
	TIME.PL	Time formatting routines
	TIME.WS	
Files to the customer:	sy\SYPRGO.GNT	Extract program parameter entry program
	sy\SYPRGU.GNT	Extract program
	SYMENU.DAT	Menu definition file
	Install.bat	Installation script
	Readme.doc	This file

Date: 3/11/2014
ISS#: ISS849

Create Enterprise Extract Program

The client was extracting information from their COLO CounterPoint system nightly to create reports using SQL scripts over an ODBC connection. The time it took to perform the extract exceeded the time window available to perform the extract. The client wanted us to write a suite of programs that allows the data to be extracted into a set of pipeline delimited files. This process needed to run considerably faster than the existing SQL/ODBC connection allows.

The client's COLO carries the data for many individual TLE stores (or locations). The data for these stores is stored as individual companies so there is a separate set of data files for each. We could not assume that the store numbers within each of the companies are unique among all of the companies. So we needed to assign each company a "Parent" store number. Using this parent store number in combination with the store numbers already assigned within each company created a unique identity for each store within the enterprise.

Setup>System>Company

To assign a parent store number to each company we added a new field "Parent company" to the company setup record. This new field is accessible as s subfield to the existing field "9. Default menu style".

The new company maintenance screen now resembles.

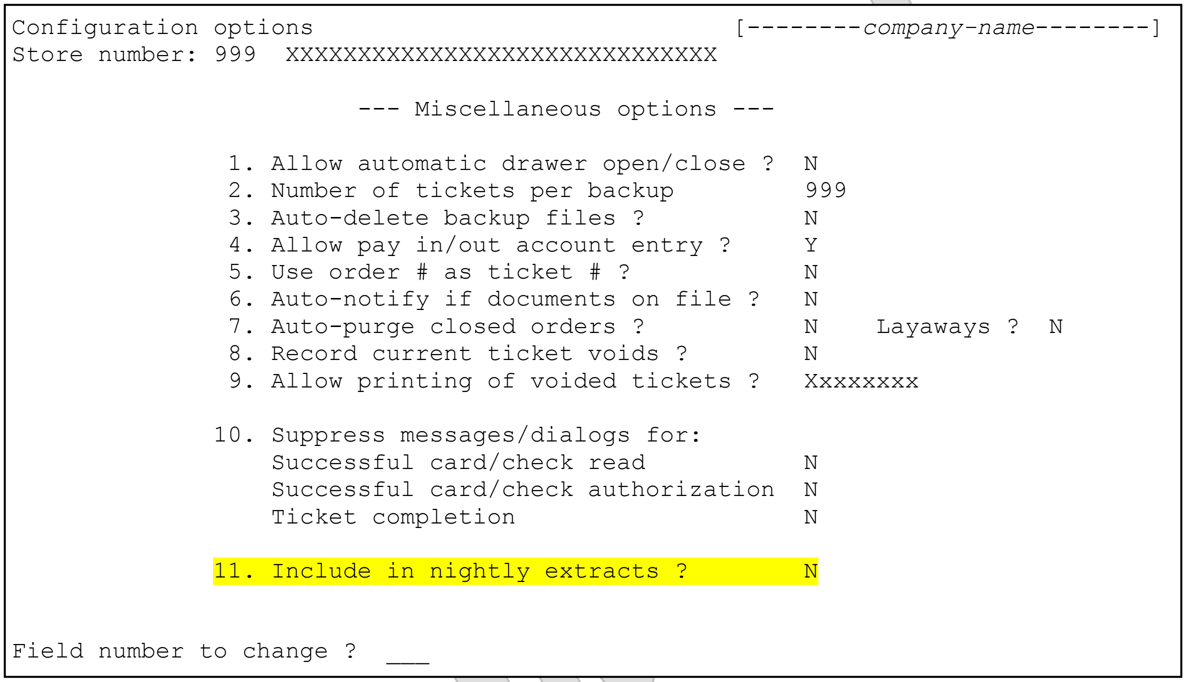
```
Company information                                [--------company-name--------]
                                                    *** Not PCI DSS Compliant ***
  1. Company report name   XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
  2. Company display name  XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
  3. Address 1             XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
  4. Address 2             XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
  5. City                  XXXXXXXXXXXXXXX  State  XX  Zip code  XXXXXXXXXX
     Use countries ?       X      Company country  XXXXXXXXXXXXXXX
  6. Phone number          XXXXXXXXXXXXXXXXXXXX
  7. Current century       9999 thru 9999
  8. Current year ID       XXXXX  99/99/99 thru 99/99/99
  9. Default menu style    Xxxxxxxx        Parent company-ID         999

 10. Allow trx for last year ?   X     17. Use sales reps ?           Y
 11. Allow trx for next year ?   X     18. Calc commissions based on  Xxxxxxxxx
 12. Account number length       9+9   19. Default commission method  Xxxxxxxxx
     Validate profit centers ?   X     20. Retain commissions ?       X
 13. Detail distributions ?      Y     21. Warn if dup rpt is within  9,999 rpts
     Print dists on journals ?   X     22. Retain payment history ?   X
 14. Allow protected changes ?   X         Use deposit management ?   X
 15. Use passwords ?             X         Retain cash detail ?       X
 16. Use workgroups ?            X     23. Restrict login exit ?      X

Field number to change ? ___
```

The new field allows a 3-digit, non-zero value.

Setup>Inventory>Locations
Setup>Point of Sale>Stores>Configuration options

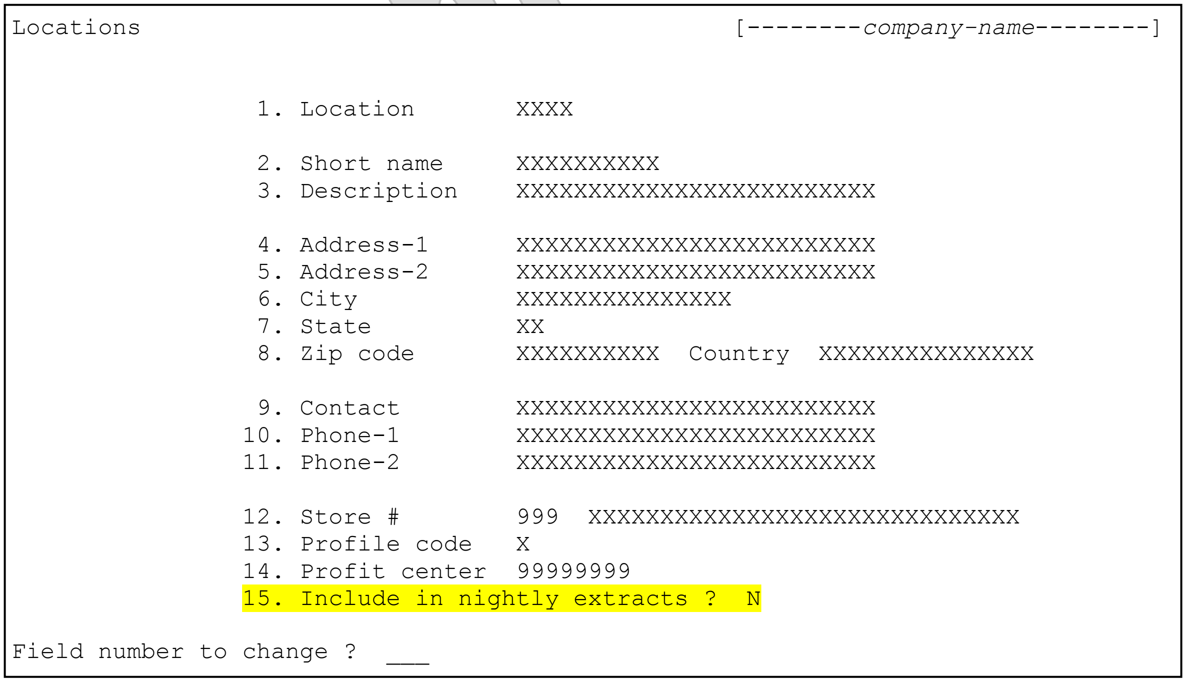
We could not assume that the client wanted to extract the information for every store and/or location within a company-id. To control which stores and locations have their information extracted we added a new Y/N field to the store and location maintenance screens. If the value in this new field is “Y” the extract program creates extract records for information relating to that store or location. If it is “N” or its value hasn’t been set, the program does not extract records for that store or location.

The store maintenance screen resembles:



```
Configuration options                             [--------company-name--------]
Store number: 999  XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

                        --- Miscellaneous options ---

               1. Allow automatic drawer open/close ?  N
               2. Number of tickets per backup         999
               3. Auto-delete backup files ?           N
               4. Allow pay in/out account entry ?     Y
               5. Use order # as ticket # ?            N
               6. Auto-notify if documents on file ?   N
               7. Auto-purge closed orders ?           N    Layaways ?  N
               8. Record current ticket voids ?        N
               9. Allow printing of voided tickets ?   Xxxxxxxx

              10. Suppress messages/dialogs for:
                  Successful card/check read           N
                  Successful card/check authorization  N
                  Ticket completion                    N

              11. Include in nightly extracts ?        N


Field number to change ?  ___
```

The location maintenance screen will resemble:



```
Locations                                         [--------company-name--------]


                 1. Location       XXXX

                 2. Short name     XXXXXXXXXX
                 3. Description    XXXXXXXXXXXXXXXXXXXXXXXXX

                 4. Address-1      XXXXXXXXXXXXXXXXXXXXXXXXX
                 5. Address-2      XXXXXXXXXXXXXXXXXXXXXXXXX
                 6. City           XXXXXXXXXXXXXXX
                 7. State          XX
                 8. Zip code       XXXXXXXXXX  Country  XXXXXXXXXXXXXXX

                 9. Contact        XXXXXXXXXXXXXXXXXXXXXXXXX
                10. Phone-1        XXXXXXXXXXXXXXXXXXXXXXXXX
                11. Phone-2        XXXXXXXXXXXXXXXXXXXXXXXXX

                12. Store #        999  XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                13. Profile code   X
                14. Profit center  99999999
                15. Include in nightly extracts ?  N

Field number to change ?  ___
```

Overview

The COLO may contain companies whose data should not be extracted (such as a demo or test company). For this reason we did not want to attempt to traverse the CounterPoint directory tree looking for datasets to process. The extraction driver program allows two methods of controlling the companies processed. The first is the command line.

The command line options resemble:

```
A XXXXX SY/SYEEXX [[/C company-id] [/P top-level-dir] [/D mm-dd-yyyy or T[-n] or A] [/I]
```

The leading parameters "A XXXXX" are required, but not used by the process. CounterPoint's own command line processing routines require these parameters be present even though our code bypasses their use. The parameter "SY/SYEXX" then follows. It is very important that this parameter is entered exactly as is is displayed here since the startup program looks for this exact parameter to know the extract program is being run and to bypass most of the normal startup routines. So spelling, case and the correct forward slash characters must be correct.

All of the remaining parameters are optional but some combinations are not allowed. Running the extract program with no optional parameters causes the program to look for and process a file named COMPEXTR.TXT (discussed below). Each parameter is discussed separately below.

/C *company-id* This parameter specifies the company-id to be processed. If this parameter is missing the program looks for and processes the file COMPEXTR.TXT (discussed later in this document). There should be a data directory with the same name as the company-id parameter. If there is only a single company defined for that installation of CounterPoint this directory might be named DATAFILS and that's the value that would be used. This parameter allows running the extraction process in an ad-hoc manner for a single company.

`/P` *`top-level-dir`* This parameter provides the path to the company's top-level directory. This is required when the company-id to be extracted is not part of the CounterPoint installation that is running the extract. Combining the value of this parameter with the *company-id* parameter (like this *top-level-dir/company-id*) should result in a valid pathspec for a directory that holds the SYCMPF.DAT file for the company being processed. This parameter is option. If it is omitted the program assumes that the company to be extracted is part of the same installation of CounterPoint that is running the extract program. This parameter can only be used when the *company-id* parameter is included on the command line.

`/D` This parameter provides the date for which records are to be extracted. A formatted date (in the format mm-dd-yyyy) or the "magic" values "T" or "T-*n*" is allowed. This parameter is optional. If it is omitted the program assumes the system date. Note that including the /D on the command line but not including either the *mm-dd-yyyy* or T parameters is the same as not having the /D parameter at all.

`[`*`mm-dd-yyyy`*`]` If a specific date value is to be extracted is should be formatted mm-dd-yyyy. Provide leading zeros for single digit month and day values (for example 02-04-2014). The program does not check that the date value is valid.

[A] This value indicates to extract all dates.

`[T[-`*`n`*`]]]` The program allows passing the "magic" values "T" or "T-*n*". "T" indicates to use the current date ("T" as in "Today"). "T-*n*" indicates to use today's date minus *n* number of days (where *n* is "1" through "9"). This will be useful if the client intends to script the program to run after midnight for the data posted the date before. Using "/D T-1" and "/D T-2" would be useful to extract the data for Saturdays and Sundays on the following Monday (when someone is around to monitor the results).

/I Master files (such as the customer and item files) don't have a date field that can be used to filter the records extracted. This could result in a large extracts. Part of CounterPoint's Multi-Site (MS) option is a flag field on records that indicate that a record has been added or modified since the last MS transmission. We'll call this the "dirty flag". Existing CounterPoint programs already maintain this flag and most files already have this field as the first field in an index which allows rapid retrieval of only the changed records. Since the client is not using MS, we used this field for our own purposes. The first time a company's data is extracted the client would probably want to extract all of the information from the master files regardless of the status of the dirty flags. Subsequent runs would extract just the records that have been added or changed.

To control this behavior the program accepts the /I parameter. If included, the program only extracts the master records that have been added or changed since the last extract. If omitted, the program extracts the entire contents of the master files. The extract program clears the dirty flag on every record that it extracts.

The COMPEXTR.TXT file

The file COMPEXTR.TXT lists the companies to be extracted and the parameters to be used with each. The client will list the CounterPoint company-id (and the parameters to be used) for each company to be processed each on its own line. (By company-id we mean the name of the top-level data file directory associated with each company such as "DATAFILS" or "DEMO1".) Within this directory should lay the file SYCMPF.DAT which the extract program will need to locate the files to be extracted. Blank lines and lines that begin with a pound sign ("#") are skipped. This allows adding comments to the file and allows temporarily commenting out an entry. Leading and trailing spaces are ignored. Any text appearing after a "#" sign is also ignored. That allows adding comments to actionable lines.

The COMPEXTR.TXT entries specify the same optional parameters already discussed for the program's command line. Command line parameters override processing the COMPEXTR.TXT file.

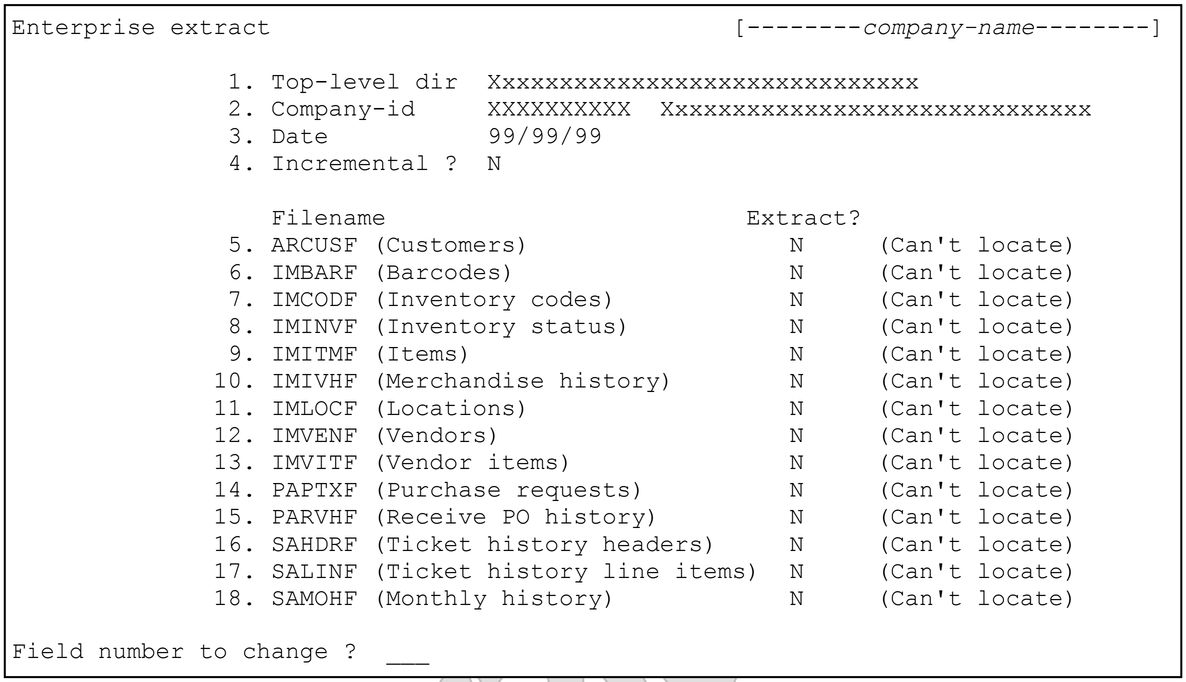
Sample COMPEXTR.TXT would be:

```
################################################################################
#                      INFINITY SOFTWARE SOLUTIONS, INC.                       #
################################################################################
#                  This file drives the enterprise extraction                  #
#                 system when the extract program (SY/SYEEXX)                  #
#              is called without parameters on the command line.               #
#          (See the file SYEEXX.BAT for samples of the command line.)          #
################################################################################
# Parameters /C company-id [/P top-level-dir] [/D mm-dd-yyyy or T[-n] or A] [/I]
#    /C company-id    The company-id whose data is to be extracted (required)
#    /P pathspec      The pathspec for the company to be extracted
#                       (optional - assumes current directory if omitted)
#    /D               The date range to use (optional - assumes today if blank)
#        mm-dd-yyyy   Specific date to use
#        A            Extract records for all dates
#        T            Use today’s date
#        T-n          Subtract n from today’s date and use that (only one digit)
#    /I   Incremental extract (optional - assumes "N" if omitted)
################################################################################
#        DO NOT ENTER ANY LINES LONGER THAN 120 CHARACTERS IN THIS FILE        #
#          Blank lines or lines with a "#" in column one are ignored           #
#               Characters following a "#" character are ignored               #
#                 Additional whitespace characters are ignored                 #
################################################################################
#                   List the companies to be extracted below                   #
#                   ----------------------------------------                   #
#             You can define an unlimited number of entries below to           #
#            extract any number of companies to be extracted.  Each            #
#            company is extracted in the order of the entries below.           #
################################################################################
# Company-id   Path           Parameters   #Optional-comments
# ----------   ------------   ----------   -----------------------
# Company-id   Path           Parameters   #Optional-comments
# ----------   ------------   ----------   -----------------------
# /C DEMO1                      # DEMO1, current install, today, non-incr
# /C DEMO1 /P c:\syn            # DEMO1 in c:\syn today, non-incr
# /C DEMO1 /P c:\syn /D A       # DEMO1 in c:\syn all dates, non-incr
# /C DEMO1 /P c:\syn /D T       # DEMO1 in c:\syn today, non-incr
# /C DEMO1 /P c:\syn /D T-1     # DEMO1 in c:\syn 1 day ago, non-incr
# /C DEMO1 /P c:\syn /D T-9     # DEMO1 in c:\syn 9 days ago , non-incr
# /C DEMO1 /I                   # DEMO1 in current install, today, incremental
# /C DEMO1 /P c:\syn /I         # DEMO1 in c:\syn today, incremental
# /C DEMO1 /P c:\syn /D A /I    # DEMO1 in c:\syn all dates, incremental
# /C DEMO1 /P c:\syn /D T /I    # DEMO1 in c:\syn today, incremental
# /C DEMO1 /P c:\syn /D T-1 /I  # DEMO1 in c:\syn 1 day ago, incremental
# /C DEMO1 /P c:\syn /D T-9 /I  # DEMO1 in c:\syn 9 days ago , incremental
 /C DEMO1 /P \syn\cp7518obj /D A      # Only this extract is allowed to run
```

File Utilities>System>Enterprise extract

The client requested a version of the extract that could be used to extract records from a specific company-id. When the extract program is called from the CounterPoint menus it presents a parameter screen. This parameter screen prompts for the command line values as well as allowing the operator to extract the information from specific files.

The parameter screen resembles:



```
Enterprise extract                                [--------company-name--------]

              1. Top-level dir  Xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
              2. Company-id     XXXXXXXXXX  Xxxxxxxxxxxxxxxxxxxxxxxxxxxxx
              3. Date           99/99/99
              4. Incremental ?  N

                 Filename                          Extract?
              5. ARCUSF (Customers)                   N     (Can't locate)
              6. IMBARF (Barcodes)                    N     (Can't locate)
              7. IMCODF (Inventory codes)             N     (Can't locate)
              8. IMINVF (Inventory status)            N     (Can't locate)
              9. IMITMF (Items)                       N     (Can't locate)
             10. IMIVHF (Merchandise history)         N     (Can't locate)
             11. IMLOCF (Locations)                   N     (Can't locate)
             12. IMVENF (Vendors)                     N     (Can't locate)
             13. IMVITF (Vendor items)                N     (Can't locate)
             14. PAPTXF (Purchase requests)           N     (Can't locate)
             15. PARVHF (Receive PO history)          N     (Can't locate)
             16. SAHDRF (Ticket history headers)      N     (Can't locate)
             17. SALINF (Ticket history line items)   N     (Can't locate)
             18. SAMOHF (Monthly history)             N     (Can't locate)

Field number to change ?  ___
```

The first two parameters can be left blank. If the first parameter is left blank the program will assume that it is extracting information from the CounterPoint installation from which the extract program is being run. If the second parameter is left blank the program will assume that it is extracting information from the currently selected CounterPoint company. Once the operator has entered the first two parameters the program will check that it can locate and open the SYCMPF.DAT file using the provided information. If it cannot the program issues an appropriate error message and stays at the second parameter.

The date parameter accepts a single date or the F1-key for “All” dates.

The parameter “4. Incremental ?” allows only a “Y” or “N” value to be entered.

Once the first two parameters have been entered the program will attempt to locate and open each file the program can extract. If the file cannot be located the literal “(Can’t locate”) will be displayed next to its parameter. If the file cannot be opened the literal “(Can’t open”) will be displayed next to its parameter. For each file that doesn’t have such an error message next to it the program will allow entering a Y/N value to indicate whether to extract the file or not.

When run in ad-hoc mode the program does not process the COMPEXTR.TXT file.

The Extract Program

We created a program that creates the extract files using the information gathered from the startup program or the parameter entry program. The actual extract program doesn't have (or need) an interface.

Processing

For each CounterPoint file processed, the program appends the extracted information to a CSV file (usually) named after the source data file with the extension CSV in a directory which is defined in the environment variable EXTRACTS. (If the environment variable is missing the program terminates with an error message.) The EXTRACTS value can be up to 30 characters long. Note that for the remainder of this document we will refer to the directory the extract files are written to as the EXTRACTS directory even though the actual directory name might be different. Each time the extract program runs it checks if the CSV file currently exists. If it does the program adds the extracted data to the existing CSV file. If the file does not exist the program creates the file, adds a header record to describe the data columns then starts adding records to the new file. This methodology allows the client to run the extract process repeatedly for any combination of companies before zipping and transmitting the data. Each time the client's transmission process consumes a CSV file it should remove the file to prevent duplicate data from being transmitted.

Backups

Once the program has completed it copies the extract files to a directory named BACKUPS located off the EXTRACTS directory. As it does this it renames the file's extension from CSV to a 3 digit number starting at "001" and incrementing with each run. This allows for up to 999 backups of 14 files – 13,986 files. That seems to be too many backups. So we implemented a rolling strategy. We allow the program to create the backup files numbered from 1 to 14. When the program finds that the first available file extension is "015" the program deletes the first 7 backup files and assigns the current backup the extension "001". Subsequent backups are named "002" through "007". While the program is testing for the next available extension to use it executes special code when checking for the extension "008". If the backup file exists and is dated older than "007", the program deletes the backup files numbered "008" through "014". The program then assigns the current backup file the extension "008". This methodology allows the client to retain at least 7, but not more than 14 backups of each file – which might still be a bit of an over-kill. Note that this process is executed for each file separately. It is possible for different extract files to be backed up using different file extension values.

Reporting

The program does not create a report. However it does create a log file containing pertinent messages such as the date and time it started and stopped, the number of record processed in each file and any errors encountered. This log file is named EXTRACTS.LOG and is located in the EXTRACTS directory. Each time the program runs it appends to this file. The extract program also logs the name of each of the backup files.

Data Formatting

The program creates a set of pipeline character ("|") delimited files. Text fields are not enclosed in double-quote characters. Double quote characters in text fields are converted to tilde characters to prevent mistranslation in whatever system uses the extracted data files. Numeric fields are formatted depending on whether the value is a date or a numeric value. Date fields are formatted mm/dd/yyyy. If the month is a single digit (January through September) the leading zero is omitted. Numeric values have the following formatting:

- No leading zeros.
- Numeric fields defined as having decimals are rounded to 2 decimal places.
- No commas, dollar signs or percent signs.
- Negative values have a leading negative sign.
- Percentage values are written so 55.43% is 55.43 (not .5543).

The records are written in the order of the file's best index available to retrieve only the desired records. So master files are in number order (item-#, customer-#...) and history files are by date (hopefully).

The program extracts information from the following files:

Filename	Description	Notes
ARCUSF	Customer File	All records are extracted
IMBARF	Barcode File	All records are extracted
IMCODF	Inventory Codes File	Cat and subcat records are extracted
IMINVF	Inventory File	All records are extracted
IMITMF	Item File	All records are extracted
IMIVHF	Merchandise History File	See notes below
IMLOCF	Location File	All records are extracted
IMVENF	Vendor File	All records are extracted
IMVITF	Vendor Item File	All records are extracted
PAPTXF	Purchase Request File	See notes below
PARVHF	Receive PO History File	See notes below
SAHDRF	Ticket Header History Header	Records posted on the date parameter are extracted
SALINF	Ticket Line History File	Records posted on the date parameter are extracted
SAMOHF	Monthly History File	See notes below

Not all records are extracted. History files only have the records that match the date parameter posted. Master files (such as the item file and customer file) do not have relevant dates and so they are extracted in their entirety. If the extract is run in incremental mode, all changed records are extracted regardless of the date range parameter.

Merchandise History File (IMIVHF)

This file contains two different records types. The first type is audit records. These records have the type “A”. They provide information at the most granular level (item/location/date/transaction-type) so there are many of them. The second are merchandise (“M”) records. These combine the activity for each item/location/date into a single record and so there are far fewer of them. We extract both types of records that match the date parameter.

Purchase Request File (PAPTXF)

This file contains the header and line item records for both purchase requests and purchase orders. The program only extracts the information for purchase orders, but it extracts all of them – no filtering by date. The client requested that we separate the headers and line items into separate files. So the program places header records in the file POHDRF.CSV and line items in POLINF.CSV.

Receive PO History File (PARVHF)

This file contains the header and line item records. The program extracts the records whose receipt date matches the date parameter. The client requested that we separate the headers and line items into separate files. So the program puts header records in the file RCVHDR.CSV and line items in RCVLIN.CSV.

Monthly History File (SAMOHF)

The monthly history file contains many different records types. Each type accumulates totals for a “calendar” month. Since the information isn’t separated by date we extract the month totals for the date parameter. The client asked that we extract the following record types from this file.

Record type	Description
AX	Customer Tax Code
CU	Customer
IN	Inventory
IT	Item
PS	Store
VE	Vendor

When processing this file for a specific date the extract program attempts to use the most efficient means to get to the desired records. To accomplish the program reads through the calendar records starting with the current records and going backwards in time until it reads the calendar record for the extraction date. It uses the prior year calendar id to get the next calendar record to check. So the client must have a continuous chain of properly linked calendar records on file going back to the oldest extraction date he intends to extract explicitly. If the extract program is run for all dates or incrementally this entire process is skipped.

File Utilities>Inventory
File Utilities>Point of Sale
File Utilities>System

Since we added a field to several records we needed to update the file utilities associated with the file's' package to support the new fields.

CounterPoint Startup Program

We modified the CounterPoint startup program to recognize the extract program on the command line and give it the ability to parse the program's command line options. We also added code to read and parse the COMPEXTR.TXT file. Between the two (the command line and the file) the program has enough information to load the linkage area and call the extract program.

CounterPoint Definition File

We added a new menu selection (*File Utilities>System>Enterprise extract*) to the CounterPoint menu definition file from which the client will run the new extract program (in ad-hoc mode).

Included with this proposal

- Modify the CounterPoint startup program as described above.
- Modify the company maintenance program as described above.
- Modify the location maintenance program as described above.
- Modify the store maintenance program as described above.
- Create the new extract parameter program as described above.
- Create the new extract program as described above.
- Modify the Inventory file utilities as described above.
- Modify the Point of Sale file utilities as described above.
- Modify the system file utilities as described above.
- Add the menu selection to both versions of the CounterPoint graphical menus.
- A simple installation procedure.
- Documentation of the enhancement and the files affected.
- A 30-day guarantee that the software operates as designed in this proposal.

Files affected by the enhancement

Source files altered:	CHKPTH.PL CHKPTH.WS	Directory checking/creation routines
	CSVRTNS.PL	CSV File creation routines
	CSVRTNS.WS	CSV File creation work fields
	IMFILU.CBL	Inventory file utilities program
	IMLOCF.FD IMLOCF.FDW IMLOCF.FDX	Location file copyfiles
	IMLOCM.CBL	Location maintenance program
	PSCFGM.CBL	Store configuration maintenance program
	PSSTRF.FD PSSTRF.FDW PSSTRF.FDX	Store file copyfiles
	SSTART.CBL	CounterPoint startup program
	SYCMPF.FD SYCMPF.FDW SYCMPF.FDX	Company file copyfiles
	SYCMPM.CBL	Company maintenance program
	SYEEXO.CBL	Extract program parameter entry program
	SYEEXX.CBL	Extract program
	SYEEXX.WS	Extract program linkage copyfile
	SYMENU.DAT	Menu definition file
	TIME.PL TIME.WS	Time formatting routines
Files to the customer:	im\IMFILU.GNT	Inventory file utilities program
	im\IMLOCM. GNT	Location maintenance program
	ps\PSCFGM. GNT	Store configuration maintenance program
	sy\SYCMPM. GNT	Company maintenance program
	sy\SYEEXO. GNT	Extract program parameter entry program
	sy\SYEEXX. GNT	Extract program
	SSTART. GNT	CounterPoint startup program
	SYMENU.DAT	Menu definition file
	SYEEXX.BAT	Extract program script
	Install.bat	Installation script
	Readme.doc	This file

Date: 4/26/2013
ISS#: ISS809

Various Tweaks

The client wanted a few changes made.

Add a Header Record to All Programs That Create an Extract File

Over the years the client has had 13 programs written (or modified) that created an extract file. Because these files were intended to be combined with the same files from the other stores they were created sans a header record to identify the columns. It turned out that the client doesn't combine the files together and so having the header record would greatly help using the files.

So we modified all of the client's modified programs that create an extract file to create a header record to identify the columns.

Revised 6-Week Inventory Report

The client reported that when they began the new year (2013) the Revised 6-Week Inventory report stopped working even though the calendars were setup correctly.

We ran the program without a problem and so turned to the source code. In stepping through the code we found that when running the program for a new year the program would issue an error message that "The report date entered must be within the current or last year's calendar". We surmised that was the problem the client was having since the operator could not proceed forward. However we found that if the operator cleared the default parameters by pressing the F1 key, the program would reload the calendar table and the desired report date could be entered. We called the client and described our finding and he confirmed that was the problem and that it had magically resolved itself. We described the problem and we both decided that sine there is an easy workaround that only needs to be performed once a year that we wouldn't need to correct the program (which would have taken quite a bit of coding).

Added User-defined Fields 3 & 4 to the Item by Sales Date and Item Sales YTD Reports

The client requested that we add total for the gross profit fields to the Item by Sales Date report.

We did.

Added User-defined Fields 3 & 4 to the Item by Sales Date and Item Sales YTD Reports

The client requested that we add user-defined fields 3 & 4 to the Item by Sales Date and Item Sales YTD reports. He wanted the fields added to the reports and the extract files the reports create.

So we did. It required reformatting the reports as follows (cut from the email that describes the changes prior to actually making them):

Item Sales by Date: The item description field is already truncated to 20 characters. But by shortening the fields "Lst-cost" and "Price-1" to a maximum value of 999.99, changing the existing user-def fields from 3 to 2 characters each and moving the fields to the left a bit, I can squeeze the new 2-character user-def fields in with no other changes.

Item Sales Year-to-Date: I'll shorten the description field from 24 to 22 characters. Then as I did in the prior report I'll shorten the fields "Lst-cost" and "Price-1" to a maximum value of 999.99, change the existing user-def fields from 3 to 2 characters each and move the fields to the left a bit. That will fit in the new 2-character user-def fields.

The new fields were both added to the right edge of the report and headered "Gr/Sb". See the samples provided at the end of this document.

Files affected by the enhancement

Source files altered:	ARCUSX.CBL	New customer birthday extract (2 versions)
	IMBPOP.CBL	Blank PO extract
	IMBSCP.CBL	Best seller by category
	IMITMX.CBL	Item data extract
	IMLSSP.CBL	Location stock status
	IMR6WP.CBL	6 Week item history
	IMRSKP.CBL	Restocking report
	IMSBDP.CBL	Item sales by date
	IMYTDP.CBL	Item sales YTD
	SACUS2.CBL	Sales by customer report (2 versions)
	SACUSX.CBL	New target customer extract (7.0.150 version)
	SAITMX.CBL	Sales data extract
	SASTRX.CBL	Benchmark data extract
Files to the customer:	ar\ARCUSX.GNT	New customer birthday extract (2 versions)
	im\IMBPOP.GNT	Blank PO extract
	im\IMBSCP.GNT	Best seller by category
	im\IMITMX.GNT	Item data extract
	im\IMLSSP.GNT	Location stock status
	im\IMR6WP.GNT	6 Week item history
	im\IMRSKP.GNT	Restocking report
	im\IMSBDP.GNT	Item sales by date
	im\IMYTDP.GNT	Item sales YTD
	sa\SACUS2.GNT	Sales by customer report
	sa\SACUSX.GNT	New target customer extract
	sa\SAITMX.GNT	Sales data extract
	sa\SASTRX.GNT	Benchmark data extract
	Readme.doc	This file

Date: 2/24/2011
ISS#: ISS678

Port Mods From Version 7.5.14 to 7.5.18

The client wanted to port the enhancements from version 7.5.14 to version 7.5.18 on both Windows and Unix.

We modified the programs as required to function within the 7.5.18 environment without any changes in the enhancement's functionality or altering newer 7.5.18 functionality.

Files affected by the enhancement

Source files altered:	ARCUSX.CBL	New customer birthday extract (2 versions)
	CHKPTH.PL	Directory checking/creation routines
	CHKPTH.WS	
	IMBPOP.CBL	Blank PO extract
	IMBSCP.CBL	Best seller by category
	IMCORU.CBL	Corporate update
	IMDRLP.CBL	Daily Restocking by location report
	IMDRRP.CBL	Daily Restocking report
	IMITMX.CBL	Item data extract
	IMLSSP.CBL	Location stock status
	IMMLSP.CBL	Multi-location stock status report
	IMR6WP.CBL	6 Week item history
	IMRSKP.CBL	Restocking report
	IMSBDP.CBL	Item sales by date
	IMYTDP.CBL	Item sales YTD
	PSEODD.CBL	POS EOD driver program
	SACUS2.CBL	Sales by customer report (2 versions)
	SACUSX.CBL	New target customer extract (7.0.150 version)
	SAITMX.CBL	Sales data extract
	SASTRX.CBL	Benchmark data extract
	SQISHL.PL	Text manipulation routines
	SQUEEZ.WS	
	SQUEZL.PL	
	SQUEZR.PL	
	SYMENU.DAT	Menu definition file

Files to the customer:	ar\ARCUSX.GNT	New customer birthday extract (2 versions)
	im\IMBPOP.GNT	Blank PO extract
	im\IMBSCP.GNT	Best seller by category
	im\IMCORU.GNT	Corporate update
	im\IMDRLP.GNT	Daily Restocking by location report
	im\IMDRRP.GNT	Daily Restocking report
	im\IMITMX.GNT	Item data extract
	im\IMLSSP.GNT	Location stock status
	im\IMMLSP.GNT	Multi-location stock status report
	im\IMR6WP.GNT	6 Week item history
	im\IMRSKP.GNT	Restocking report
	im\IMSBDP.GNT	Item sales by date
	im\IMYTDP.GNT	Item sales YTD
	ps\PSEODD.GNT	POS EOD driver program
	sa\SACUS2.GNT	Sales by customer report
	sa\SACUSX.GNT	New target customer extract
	sa\SAITMX.GNT	Sales data extract
	sa\SASTRX.GNT	Benchmark data extract
	SYMENU.DAT	Menu definition file
	INSTALL1.BAT	Installation script
	INSTALL2.BAT	Installation script
	Readme.doc	This file

Date: 6/1/2009
ISS#: ISS553

New Daily Restocking by Location Report

The client wanted a modified version of the existing Daily Restocking report that is filtered by locations (rather than store-#s) and had a different report format.

We created the new Daily Restocking by Location report from the existing Daily Restocking report. To better discern this new report from the existing report we also renamed the existing report Daily Restocking by Store report.

The new report has the same parameter screen with the exception of the second parameter which we changed from the entry of a single store-# value to allow entering up to 5 locations.

The new parameter screen will resemble:

```
Daily restocking by location report                [--------company-name--------]



      Please enter:

           1. Print in order by  Xxxxxxxxxxxxx

           2. Locations          XXXX  XXXX  XXXX  XXXX  XXXX

           3. Vendor #           XXXXXXXXXXXX  Xxxxxxxxxxxxxxxxxxxxxxxxx

           4. Categ/Subcat       XXXXX/XXXXX

           5. Time range         99:99xxxxx to 99:99xxx



Field number to change ?  __
```

1. Print in order by	Allows “Vendor/Item” or “Category/Item”
2. Locations	Allow F2-key to toggle between Locations and Location-groups Allow F9 key for DataLookup
3. Vendor #	Allows F1 key for “All” Allow F9 key for DataLookup
4. Categ/Subcat	Allow F1 keys for “All” on both fields Allow F9 key for DataLookup
5. Time range	Allow F1 keys for “Earliest” to “Latest”

If the operator enters a location group for parameter 2, the program retrieves the locations assigned to the location (up to five) and reports them as if the locations had been keyed separately. Only a single location group value is allowed when entering the parameters.

The report includes all inventory type items (there is no filter for item-#s). The sales column reports the sales for the entered timeframe during the current system date. The values come from the unposted tickets in POS (as opposed to posted tickets in the ticket history file). The Avail column display the item’s calculated quantity-available (at all locations). The YTD sales column values is the sales found in the unposted tickets added with sales retrieved from ticket history. The location column values (on the right side of the report) are the quantity available calculated from each item’s current inventory records.

The report headers display the locations entered in the order they are entered (or entered into the location group record). The report does not have any control breaks or subtotals. The only totals are for the total number of items reported and the total sales of all items on the current system date. The format of the report does not change regardless of the sort order. The detail lines merely appear in the desired sequence. Unlike the original Daily Restocking report, this report prints single spaced (not double spaced).

The report has the columns:

Item-#	From the item record
Description	From the item record
Retail	Regular price from the item record
Sold	Accumulated from the POS ticket files
Available	Calculated from the item record – QtyOnHand minus QtyCommitted
OnOrd	From the inventory record
YTD	Accumulated from the ticket history files plus the current quantity-sold
Category and Subcategory	From the item record
Other location OHs	Calculated from the each inventory record – QtyOnHand minus QtyCommitted

See the end of this document for an example of the resulting report.

<u>Included with this proposal</u>

- Rename the existing Daily Restocking report as described above.
- Create the new Daily Restocking by Location report as described above.
- A simple installation procedure.
- Documentation of the enhancement and the files affected.
- A 30-day guarantee that the software operates as designed in this proposal.

<u>Files affected by the enhancement</u>

Source files altered:	IMDRLP.CBL	Daily Restocking by location report
	IMDRRP.CBL	Daily Restocking by store report
	SYMENU.DAT	Menu definition file
Files to the customer:	im\IMDRLP.GNT	Daily Restocking by location report
	im\IMDRRP.GNT	Daily Restocking by store report
	SYMENU.DAT	Menu definition file
	Readme.doc	This file

Date: 2/11/2009
ISS#: ISS519 (continued)

Change the Extract Parameters on Select Reports

In the last round of mods we modified the way user control which extract files get created. The client discovered that one of the programs (the Sales by customer report) would consistently create an empty extract file if the operator specified to create a local file only.

We tested it found it to be true and corrected the problem.

Files affected by the enhancement

Source files altered:	SACUS2.CBL	Sales by customer report (2 versions)
Files to the customer:	sa\SACUS2.GNT	Sales by customer report
	Readme.doc	This file

Date: 1/26/2009
ISS#: ISS519

Change the Extract Parameters on Select Reports

In the last round of mods we created the new Multi-Location Stock Status report. This report differed in the way we have been controlling the extraction of the data files from the way all of the other extract programs operate. The client wanted us to change certain other extract program to operate in the same manner.

The new methodology replaced the existing parameter with the parameters:

```
14. Send data file to Corp ?  X       Store-#  999
15. Save data file locally ?  X      Filename  Xxxxxxxxxxxx
```

The client had already defined values for the environment variables EXTRACT and CSVDATA. EXTRACT holds the path to the corporate extract file directory. CSVDATA holds the path to the store's local extract file directory. Both variables contain only the path – not the filenames.

If the operator replies "N" to "14. Send data file to Corp ?", the program does not create/copy the extract file to the corporate path (EXTRACTS). If the operator answers "Y", the program creates/copies the file to the path stored in the EXTRACTS environment variable. The program then prompts for a subparameter called "Store-#". The value entered into this field must be greater than zeros. However the system does not verify the value entered into this field. This store-# is appended to the program's name to create the corporate extract's filename.

If the operator replies "N" to "15. Save data file locally ?", the program does not create/copy the extract file to the local path (CSVDATA). If the operator answers "Y", the program creates/copies the file to the path stored in the CSVDATA environment variable. If the operator answers "N" to parameter "14. Send data file to Corp ?", the program prompts for the local file's filename. This filename is limited to 8.3 format. If the operator answers "Y" to parameter "14. Send data file to Corp ?", the program does not prompt for the local file's filename. Instead it uses the corporate filename as the local filename as well.

Note that if an extract file is to be created it may be created in the path stored in CSVDATA or EXTRACTS (or both). Rather than create the extract file in a temporary file the program creates the file in desired folder with preference to the path in the CSVDATA variable. So if the operator indicates to create a corporate extract file, but not a local extract file, the program creates the extract file in the path specified in the EXTRACTS variable. If the operator indicates to create a local extract file, the program creates the extract file in the directory from the CSVDATA variable. If the operator also indicated to create a corporate extract file, the program copies the extract file from the local directory to the corporate directory once it has been completed.

Other than the extract file parameters, the rest of each program's functionality remains as-is.

We changed the following programs to control the extraction of their files in the same manner.

Title	Program file
Item sales year-to-date	IMYTDP.GNT
Item sales by date	IMSBDP.GNT
Best seller by category	IMBSCP.GNT
Sales analysis by customer	SACUS2.GNT
Target customer extract	SACUSX.GNT

Included with this enhancement

- Modify the programs listed as described above.
- A simple installation procedure.
- Documentation of the enhancement and the files affected.
- A 30-day guarantee that the software operates as designed in this proposal.

Files affected by the enhancement

Source files altered:	IMBSCP.CBL	Best seller by category
	IMSBDP.CBL	Item sales by date
	IMYTDP.CBL	Item sales YTD
	SACUS2.CBL	Sales by customer report (2 versions)
	SACUSX.CBL	New target customer extract (7.0.150 version)
Files to the customer:	im\IMBSCP.GNT	Best seller by category
	im\IMSBDP.GNT	Item sales by date
	im\IMYTDP.GNT	Item sales YTD
	sa\SACUS2.GNT	Sales by customer report
	sa\SACUSX.GNT	New target customer extract
	Readme.doc	This file

Date: 12/2/2008
ISS#: ISS499 (continued)

New Multi-Location Stock Status Report

Several stores reported the sales column was reporting whacked information. We agreed and fixed it.

Files affected by the enhancement

Source files altered:	IMMLSP.CBL	Multi-location stock status report
Files to the customer:	im\IMMLSP.GNT	Multi-location stock status report
	Readme.doc	This file

Date: 11/4/2008
ISS#: ISS499 (continued)

New Multi-Location Stock Status Report

That last fix corrected only the parts I tested when in fact the problem was considerably wider in scope. It turns out the original Location Stock Status report handled all of the sales and quantity parameters incorrectly.

So I went back through the code and (hopefully) changed every place that did it wrong.

Files affected by the enhancement

Source files altered:	IMMLSP.CBL	Multi-location stock status report
Files to the customer:	im\IMMLSP.GNT	Multi-location stock status report
	Readme.doc	This file

Date: 10/20/2008
ISS#: ISS499 (continued)

New Multi-Location Stock Status Report

In testing, the client discovered that the program was not applying the report parameters correctly. The client wanted the filtering parameters (for sales amount and quantities) to be applied to the first location only. The information for the subsequent locations was to be reported regardless of whether their values passed the filters.

We didn't write the program this new program was created from and we didn't touch this part of the code when we used the original to create the new report. However, we did fix it for the client since it needed it. We never saw the original specs for the program so we couldn't possibly have known there was anything wrong.

Files affected by the enhancement

Source files altered:	IMMLSP.CBL	Multi-location stock status report
Files to the customer:	im\IMMLSP.GNT	Multi-location stock status report
	Readme.doc	This file

Date: 10/14/2008
ISS#: ISS499

New Multi-Location Stock Status Report

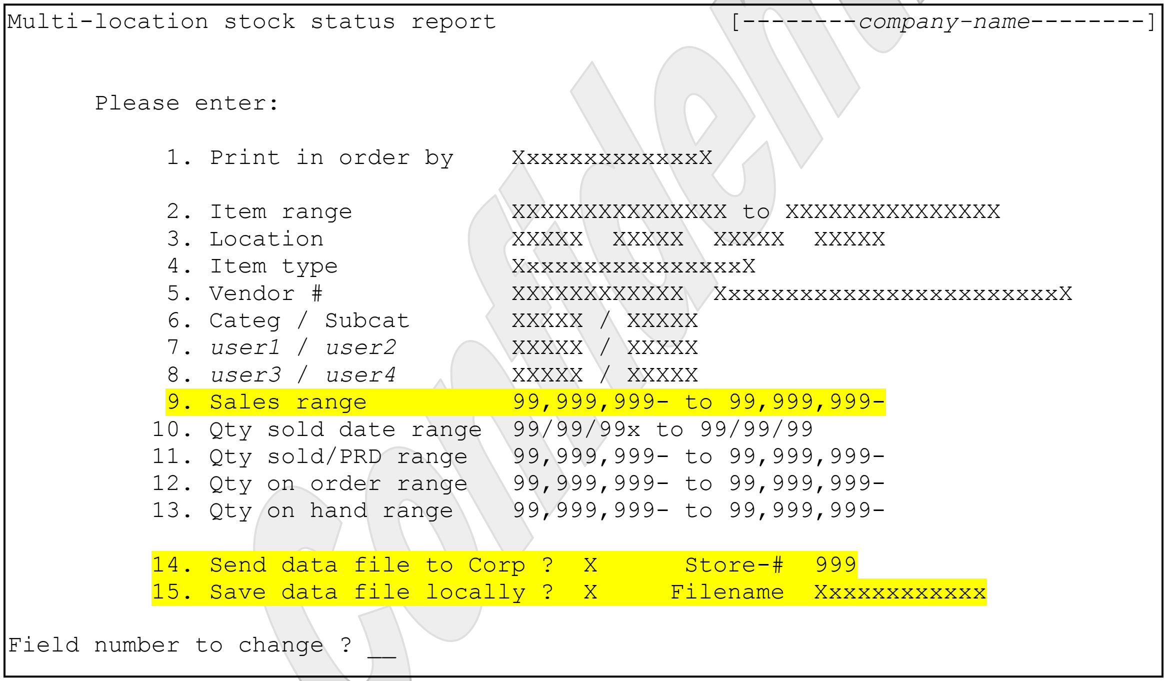
The client wanted modifications made to the existing Stock Status report to create a new report named the Multi-Location Stock Status report. The new report has a new filter parameter (named “Sales range”). It also has substantial changes to the way the operator controls the extract files that are created.

The existing report allowed entering up to 4 location codes or location groups (or a mix of both). The operator did not have to utilize all four fields. We did not change this functionality. However we did change the program so that it does not populate the header fields for any unused location/group columns.

We inserted the new parameter as parameter number nine. Any items whose totals sales within the “Qty sold date range” do not fall within the values of this parameter do not appear on the report.

We changed the old parameter “Create CSV file” to a pair of parameters. One parameter controls the generation of the corporate extract file and the other controls the generation of the local extract file. The two parameters operate independent of the other.

The new parameter screen resembles:



```
Multi-location stock status report                [--------company-name--------]


      Please enter:

           1. Print in order by    XxxxxxxxxxxxxX

           2. Item range           XXXXXXXXXXXXXXX to XXXXXXXXXXXXXXX
           3. Location             XXXXX  XXXXX  XXXXX  XXXXX
           4. Item type            XxxxxxxxxxxxxxxX
           5. Vendor #             XXXXXXXXXXXX  XxxxxxxxxxxxxxxxxxxxxxxxX
           6. Categ / Subcat       XXXXX / XXXXX
           7. user1 / user2        XXXXX / XXXXX
           8. user3 / user4        XXXXX / XXXXX
           9. Sales range          99,999,999- to 99,999,999-
          10. Qty sold date range  99/99/99x to 99/99/99
          11. Qty sold/PRD range   99,999,999- to 99,999,999-
          12. Qty on order range   99,999,999- to 99,999,999-
          13. Qty on hand range    99,999,999- to 99,999,999-

          14. Send data file to Corp ?  X       Store-#  999
          15. Save data file locally ?  X      Filename  Xxxxxxxxxxxx

Field number to change ? __
```

The client has already defined values for the environment variables EXTRACT and CSVDATA. EXTRACT holds the path to the corporate extract file directory. CSVDATA holds the path to the store’s local extract file directory. Both variables contain only the path – not the filenames.

If the operator replies “N” to “14. Send data file to Corp ?”, the program does not create/copy the extract file to the corporate path (EXTRACTS). If the operator answers “Y”, the program creates/copies the file to the path stored in the EXTRACTS environment variable. The program then prompts for a subparameter called “Store-#”. The value entered into this field must be greater than zeros. However the system does not verify the value entered into this field. This store-# is appended to the program’s name (IMMLSP) to create the corporate extract’s filename.

If the operator replies “N” to “15. Save data file locally ?”, the program does not create/copy the extract file to the local path (CSVDATA). If the operator answers “Y”, the program creates/copies the file to the path stored in the CSVDATA environment variable. If the operator answers “N” to parameter “14. Send data file to Corp ?”, the program prompts for the local file’s filename. This filename is limited to 8.3 format. If the operator answers “Y” to parameter “14. Send data file to Corp ?”, the program does not prompt for the local file’s filename. Instead it uses the corporate filename as the local filename as well.

Note that if an extract file is to be created it may be created in the path stored in CSVDATA or EXTRACTS (or both). Rather than create the extract file in a temporary file the program creates the file in desired folder with preference to the path in the CSVDATA variable. So if the operator indicates to create a corporate extract file, but not a local extract file, the program creates the extract file in the path specified in the EXTRACTS variable. If the operator indicates to create a local extract file, the program creates the extract file in the directory from the CSVDATA variable. If the operator also indicated to create a corporate extract file, the program copies the extract file from the local directory to the corporate directory once it has been completed.

We made several changes to the report's format:

- If the operator specifies three or less locations the report is printed in one-line format. If the operator specifies four locations the program splits the item information (the fields on the left side of the report) onto a separate line and prints the location information on its own line. See the end of this document for a sample of the resulting formats.

- We changed the format of location headers. If a location is being reported, the literal is "|-----Loc:XXXX------|". If a location group is being reported the program displays the literal "|----Grp:XXXXX------|". The program used to display "LOCATION XXXXX" for both.

- We added 2 new columns to the existing report's format. These new columns carry the item's currently assigned subcategory and class. (The field class is a user-defined field assigned the tag "Class".) These fields were inserted between the existing fields "Prc-1" and "Pak". Each field is truncated to the first three characters.

- We renamed the field "Prc-1" to "Price" and changed the value printed from the item's price-1 field to the item's regular-price field.

- We renamed the column "L-cost" (last-cost) to "Cost".

- We expanded the quantity fields from three digits to four digits (without a thousands separator).

- We replaced the field "Qdt" (quantity-sold in report period) with "Sales" (the amount of sales within the report period). This value comes from the ticket history files. We expanded this field from three digits to six digits (without a thousands separator). We rounded the sales amount to the nearest whole dollar amount.

- We replaced the field "Qyd" (quantity-sold year-to-date) with "Qty" (the quantity sold within the report period). The source for this field changed from the monthly history file to the ticket history file.

- We renamed the column "Qoh" (quantity-on-hand) to "OH".

- We renamed the column "Qoo" (quantity-on-order) to "OO"

With the exception of the field "Sales" (which is six digits), each of the four fields being reported for each location is only four digits in length. If the value being loaded into one of these 4-digit fields is greater than 9999 the program loads the literal "^^^^" into the printed field. If the value to be loaded into one of the 4-digit fields is less than -9999, the program loads the literal "vvvv" into the printed field. The program was loading the literal "999" in the printed field in either case.

If the value being loaded into the field "Sales" is greater than 999999 the program loads the literal "^^^^^^" into the printed field. If the value to be loaded into the field "Sales" is less than -999999, the program loads the literal "vvvvvv" into the printed field.

Note that with the addition of the new fields the one-line formatted report became wider than CounterPoint's normal 132 columns. In order to bring the column count within limits we truncated the item description field 2 characters to 23 characters in length. Since the two-line format has the item information printed on a separate line it is not necessary to truncate the description field for that format.

See the end of this document for an example of the resulting report.

We also made some changes to the format of the extract file(s).

- We added a header row to the file. The field names used match the column names on the report. The four field names for the location columns repeat, each being appended with the column's corresponding location/group value to separate and identify the four sets of column values.

- We changed the format of negative numbers to have a leading hyphen.

- We included the new columns in the data rows in the same position in which they appear in the report. Basically, each field on the report is written to the extract record in the order they appear on the report.

<u>Included with this enhancement</u>

- Create the new Multi-Location Stock Status report as described above.
- A simple installation procedure.
- Documentation of the enhancement and the files affected.
- A 30-day guarantee that the software operates as designed in this proposal.

<u>Files affected by the enhancement</u>

Source files altered:	IMMLSP.CBL	Multi-location stock status report
Files to the customer:	im\IMMLSP.GNT	Multi-location stock status report
	Readme.doc	This file

Date: 11/19/2007
ISS#: ISS411 (continued)

Add Store-# Parameter to New Daily Restocking Report
The client discovered that FNTC wouldn't provide access to the last parameter in the Daily Restocking report.

We corrected that problem (existed since we added the store-# parameter on 9/6/2007). We also corrected a minor flaw that allowed the operator to press the F1 key at the store-# parameter although the parameter should not have allowed it. (Existed since the program was initially written.)

Files affected by the enhancement

Source files altered:	IMDRRP.CBL	Daily Restocking report
Files to the customer:	im\IMDRRP.GNT	Daily Restocking report
	Readme.doc	This file

Date: 10/23/2007
ISS#: ISS426

Port Mods From Version 7.5.10 to 7.5.14

The client wanted to port the enhancements from version 7.5.10 to version 7.5.14 on both Windows and Unix.

We modified the programs as required to function within the 7.5.14 environment without any changes in the enhancement's functionality or altering newer 7.5.14 functionality.

Files affected by the enhancement

Source files altered:	ARCUSX.CBL	New customer birthday extract (2 versions)
	CHKPTH.PL	Directory checking/creation routines
	CHKPTH.WS	
	IMBPOP.CBL	Blank PO extract
	IMBSCP.CBL	Best seller by category
	IMCORU.CBL	Corporate update
	IMDRRP.CBL	Daily Restocking report
	IMITMX.CBL	Item data extract
	IMLSSP.CBL	Location stock status
	IMR6WP.CBL	6 Week item history
	IMRSKP.CBL	Restocking report
	IMSBDP.CBL	Item sales by date
	IMYTDP.CBL	Item sales YTD
	PSEODD.CBL	POS EOD driver program
	SACUS2.CBL	Sales by customer report (2 versions)
	SACUSX.CBL	New target customer extract (7.0.150 version)
	SAITMX.CBL	Sales data extract
	SASTRX.CBL	Benchmark data extract
	SQISHL.PL	Text manipulation routines
	SQUEEZ.WS	
	SQUEZL.PL	
	SQUEZR.PL	
	SYMENU.DAT	Menu definition file

Files to the customer:	ar\ARCUSX.GNT	New customer birthday extract (2 versions)
	im\IMBPOP.GNT	Blank PO extract
	im\IMBSCP.GNT	Best seller by category
	im\IMCORU.GNT	Corporate update
	im\IMDRRP.GNT	Daily Restocking report
	im\IMITMX.GNT	Item data extract
	im\IMLSSP.GNT	Location stock status
	im\IMR6WP.GNT	6 Week item history
	im\IMRSKP.GNT	Restocking report
	im\IMSBDP.GNT	Item sales by date
	im\IMYTDP.GNT	Item sales YTD
	ps\PSEODD.GNT	POS EOD driver program
	sa\SACUS2.GNT	Sales by customer report
	sa\SACUSX.GNT	New target customer extract
	sa\SAITMX.GNT	Sales data extract
	sa\SASTRX.GNT	Benchmark data extract
	SYMENU.DAT	Menu definition file
	INSTALL1.BAT	Installation script
	INSTALL2.BAT	Installation script
	Readme.doc	This file

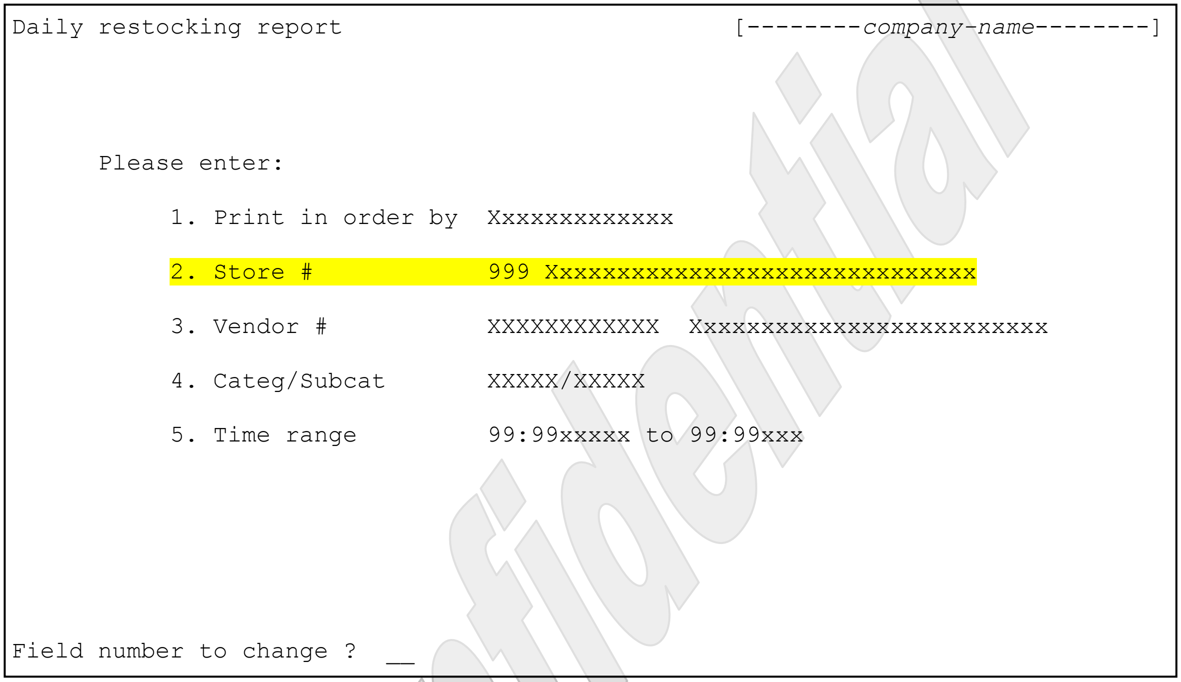
Date: 9/6/2007
ISS#: ISS411

Add Store-# Parameter to New Daily Restocking Report

The client requested we add a new store-# parameter to the Restocking report we created for them in May.

The values printed on the report reflect the inventory values for the location associated with value of the store parameter.

The new parameter screen resembles:



```
Daily restocking report                           [--------company-name--------]



      Please enter:

           1. Print in order by  Xxxxxxxxxxxxx

           2. Store #            999 Xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

           3. Vendor #           XXXXXXXXXXXX  Xxxxxxxxxxxxxxxxxxxxxxxxx

           4. Categ/Subcat       XXXXX/XXXXX

           5. Time range         99:99xxxxx to 99:99xxx




Field number to change ?  __
```

Files affected by the enhancement

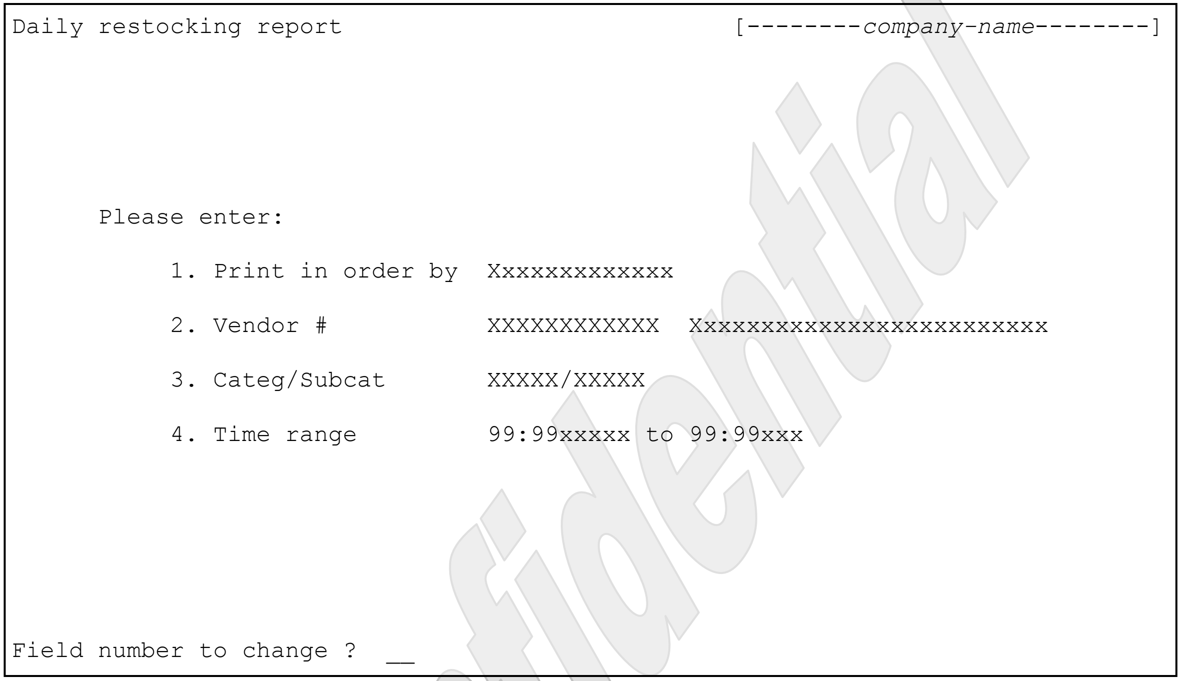
Source files altered:	IMDRRP.CBL	Daily Restocking report
Files to the customer:	im\IMDRRP.GNT	Daily Restocking report
	Readme.doc	This file

Date: 5/21/2007
ISS#: ISS385

Create New Daily Restocking Report

The client requested a new report based off the existing Restocking report that had fewer parameters and reported only items that had been sold on the current date. The report does not look at the history files (except to calculate the YTD amount). It only reports sales from the POS ticket files and only tickets dated on the current date.

The parameter screen resembles:



```
Daily restocking report                           [--------company-name--------]




      Please enter:

           1. Print in order by  Xxxxxxxxxxxxx

           2. Vendor #           XXXXXXXXXXXX  Xxxxxxxxxxxxxxxxxxxxxxxxx

           3. Categ/Subcat       XXXXX/XXXXX

           4. Time range         99:99xxxxx to 99:99xxx




Field number to change ?  __
```

`1. Print in order by`	`Allows` "Vendor/Item" or "Category/Item"
`2. Vendor #`	Allows F1 key for "All" Allow F9 key for DataLookup
`3. Categ/Subcat`	Allow F1 keys for "All" on both fields Allow F9 key for DataLookup
`4. Time range`	Allow F1 keys for "Earliest" to "Latest"

The report includes all inventory-type items. The program retrieves the store configuration record for the current location and uses the default location assigned to the store to filter the items purchased.

The report has the columns:

Item-#	From the item record
Description	From the item record
Status	From the item record
Sold	Accumulated from the POS ticket files
Available	Calculated from the inventory record – QtyAvailable minus QtyCommitted
Xfer	From the inventory record
OnOrd	From the inventory record
QtyYTD	Accumulated from the ticket history files plus the current quantity-sold
Vendor	From the item record
Category and Subcategory	From the item record

See the end of this document for a sample of the report.

<u>Files affected by the enhancement</u>

Source files altered:	IMDRRP.CBL	Daily Restocking report
Files to the customer:	im\IMDRRP.GNT	Daily Restocking report
	Readme.doc	This file

Date: 11/21/2006
ISS#: ISS339 (continued)

Port Mods From Version 7.5.8 to 7.5.10 With Changes
More tweaks.

The server host was having problems with the extracts for a couple of new sites. There were a couple of issues that were raised when they changed the remark characters in the STOREMAP.TXT file from semicolons to pound signs. Doing so allowed the programs to assume the pound sign records were valid mapping records which tricked the programs into trying to use these records as mapping records. Secondly if the program read through the entire STOREMAP file and did not find a matching record the file status was "AT-END". The program then tested the file status and assumed that now valid records were found even if one was (the first valid record on file even if it was not a match). Evidentially the stores thus far have STOREMAP.TXT files that contain entries that match the CounterPoint company's store-#.

So we added the pound-sign character as a remark character for the STOREMAP.TXT file (in addition to the semicolon character). We also changed the extract programs not to test the file status of the STOREMAP.TXT file after attempting to get the correct mapping record. Instead we are only looking to see that we either got a matching record or at least one valid record.

Files affected by the enhancement

Source files altered:	ARCUSX.CBL	New customer birthday extract (2 versions)
	IMITMX.CBL	Item data extract
	SACUS2.CBL	Sales by customer report (2 versions)
	SACUSX.CBL	New target customer extract (7.0.150 version)
	SAITMX.CBL	Sales data extract
	SASTRX.CBL	Benchmark data extract
Files to the customer:	ar\ARCUSX.GNT	New customer birthday extract (2 versions)
	im\IMITMX.GNT	Item data extract
	sa\ SACUS2.GNT	Sales by customer report
	sa\ SACUSX.GNT	New target customer extract
	sa\SAITMX.GNT	Sales data extract
	sa\SASTRX.GNT	Benchmark data extract
	Readme.doc	This file

Date: 10/24/2006
ISS#: ISS339 (continued)

Port Mods From Version 7.5.8 to 7.5.10 With Changes
Once the client's host site installed the mods we discovered that some tweaks were in order.

The host is copying the files to a path that is completely separated from the CounterPoint directory. The routine that we used to check that this directory exists would disassemble the CSVDATA path checking that each part of the path existed. Since the host does not allow write access to the root of the drive the programs would fail to load the CSVDATA value. We changed the programs to first check that the CSVDATA directory exists in its entirety first, before trying to create the individual parts of the path. This will correct the problem the host is having however it will also require the CSVDATA path to be set up manually (on Unix / Linux systems only) before the extracts will run.

In the last round of mods we started using the CSVDATA environment variable to point to the directory in which the extract files should be copied. The programs were not writing the extract files to "private" directories before copying them to the CSVDATA path. This update introduces a new environment variable EXTRACTS to the system. Each of the extract programs now create the extract file in the directory specified by the value in the EXTRACTS environment variable. The file is then copied to the directory specified by the CSVDATA environment variable as before.

Also, all extract files are now handled the same way. Previously corporate extract files were created in a different directory than non-corporate extract files. Some extracts were created with TXT file extensions and some used mapped values or values keyed as a parameter. We now use either the keyed parameter value or the mapped value only. Extract files are no longer created with TXT or CSV extensions.

Files affected by the enhancement

Source files altered:	ARCUSX.CBL	New customer birthday extract (2 versions)
	CHKPTH.PL	Directory checking/creation routines
	IMBPOP.CBL	Blank PO extract
	IMBSCP.CBL	Best seller by category
	IMITMX.CBL	Item data extract
	IMLSSP.CBL	Location stock status
	IMR6WP.CBL	6 Week item history
	IMRSKP.CBL	Restocking report
	IMSBDP.CBL	Item sales by date
	IMYTDP.CBL	Item sales YTD
	SACUS2.CBL	Sales by customer report (2 versions)
	SACUSX.CBL	New target customer extract (7.0.150 version)
	SAITMX.CBL	Sales data extract
	SASTRX.CBL	Benchmark data extract
Files to the customer:	ar\ARCUSX.GNT	New customer birthday extract (2 versions)
	im\IMBPOP.GNT	Blank PO extract
	im\IMBSCP.GNT	Best seller by category
	im\IMITMX.GNT	Item data extract
	im\IMLSSP.GNT	Location stock status
	im\IMR6WP.GNT	6 Week item history
	im\IMRSKP.GNT	Restocking report
	im\IMSBDP.GNT	Item sales by date
	im\IMYTDP.GNT	Item sales YTD
	sa\SACUS2.GNT	Sales by customer report
	sa\SACUSX.GNT	New target customer extract
	sa\SAITMX.GNT	Sales data extract
	sa\SASTRX.GNT	Benchmark data extract
	Readme.doc	This file

Date: 9/26/2006
ISS#: ISS339

Port Mods From Version 7.5.8 to 7.5.10 With Changes

The client wanted to port the enhancements from version 7.5.8 to version 7.5.10 on both Windows and Unix. At the same time he wanted to make a few changes:

- Use the corporate store number as the extracted file's extension.
- Change the location the extracted files were written to. Corporate extracts and regular extracts were written to separate directories.
- Add code to copy the corporate extract files to a separate directory in preparation for them to be transmitted.
- Change the location of the CORPPATH, LOCMAP and STOREMAP files.

The client defined the corporate extract files as being programs IMITMX (ITEMS.999), IMYTDP (IMYTDP.999), SAITMX (SALES.999) and SASTRX (STATS.999). All other programs are considered regular extract programs.

We modified the corporate extract programs in this manner:

- We added code to check for and create the directory CORPEXTR when the program first starts. If there is an error creating the directory the program displays an appropriate error message and terminates.
- We added code to retrieve the contents of the environment variable CSVDATA. This variable should contain the path the extract file should be copied to once the extract has completed. If the variable has not been defined the program displays an appropriate error message and terminates. The program then creates the directory (or directories if none of the path exists). The definition of CSVDATA should be added to the file SYNRUN.SYN which is loaded at CounterPoint's startup.
- The corporate extract programs now create their extract file in the directory CORPEXTR.
- The extract filename's extension has been changed to reflect the corporate store-# for the location running the extract. One of the corporate extract programs allow this value to be entered as a parameter. The others use the store-mapping file (STOREMAP.TXT) to translate the current store-# to the corporate store-#. The programs read through the store-mapping file looking for the first record that matches the store-# the operator has been assigned to. So if the operator is assigned to store-# "002" the program will read through the store-mapping file looking for the first record that maps store-# "002" to a corporate store-#. As it does this it retains the first valid corporate store-# it finds in the file (even if it is not the store-# assigned to the operator). If the program finds the correct record for the operator's store-# it uses the mapped corporate store-#. Otherwise it will use the first corporate store-# it found. If the program didn't find the matching store-mapping record and did not find a first valid corporate store-# record it displays an appropriate error message and the program terminates.
- Once the extract file has been created the program copies the extract file from the CORPEXTR directory to the directory assigned to the environment variable CSVDATA.

We modified the regular extract programs in this manner:

- We added code to check for and create the directory EXTRACTS when the program first starts. If there is an error creating the directory the program displays an appropriate error message and terminates.
- The corporate extract programs now create their extract file in the directory EXTRACTS.
- The extract filename's extension has been changed to reflect the corporate store-# for the location running the extract. Most of the regular extract programs allow this value to be entered as a parameter. The others use the store-mapping file (STOREMAP.TXT) to translate the current store-# to the corporate store-#. The programs read through the store-mapping file looking for the first record that matches the store-# the operator has been assigned to. So if the operator is assigned to store-# "002" the program will read through the store-mapping file looking for the first record that maps store-# "002" to a corporate store-#. As it does this it retains the first valid corporate store-# it finds in the file (even if it is not the store-# assigned to the operator). If the program finds the correct record for the operator's store-# it uses the mapped corporate store-#. Otherwise it will use the first corporate store-# it found. If the program didn't find the matching store-mapping record and did not find a first valid corporate store-# record it displays an appropriate error message and the program terminates.

No other changes were made to the client's enhancements.

Files affected by the enhancement

Source files altered:	ARCUSX.CBL	New customer birthday extract (2 versions)
	CHKPTH.PL	Directory checking/creation routines
	CHKPTH.WS	
	IMBPOP.CBL	Blank PO extract
	IMBSCP.CBL	Best seller by category
	IMCORU.CBL	Corporate update
	IMITMX.CBL	Item data extract
	IMLSSP.CBL	Location stock status
	IMR6WP.CBL	6 Week item history
	IMRSKP.CBL	Restocking report
	IMSBDP .CBL	Item sales by date
	IMYTDP .CBL	Item sales YTD
	PSEODD.CBL	POS EOD driver program
	SACUS2.CBL	Sales by customer report (2 versions)
	SACUSX.CBL	New target customer extract (7.0.150 version)
	SAITMX.CBL	Sales data extract
	SASTRX.CBL	Benchmark data extract
	SQISHL.PL	Text manipulation routines
	SQUEEZ.WS	
	SQUEZL.PL	
	SQUEZR.PL	
	SYMENU.DAT	Menu definition file
Files to the customer:	ar\ARCUSX.GNT	New customer birthday extract (2 versions)
	im\IMBPOP.GNT	Blank PO extract
	im\IMBSCP.GNT	Best seller by category
	im\IMCORU.GNT	Corporate update
	im\IMITMX.GNT	Item data extract
	im\IMLSSP.GNT	Location stock status
	im\IMR6WP.GNT	6 Week item history
	im\IMRSKP.GNT	Restocking report
	im\IMSBDP .GNT	Item sales by date
	im\IMYTDP .GNT	Item sales YTD
	ps\PSEODD.GNT	POS EOD driver program
	sa\ SACUS2.GNT	Sales by customer report
	sa\ SACUSX.GNT	New target customer extract
	sa\SAITMX.GNT	Sales data extract
	sa\SASTRX.GNT	Benchmark data extract
	SYMENU.DAT	Menu definition file
	INSTALL1.BAT	Installation script
	INSTALL2.BAT	Installation script
	Readme.doc	This file

Date: 5/25/2005

ISS#: ISS221 (continued)

<u>Changes to Extract Programs</u>

When the Benchmark or Sales extracts were run from the menus (the operator entered the reporting date) the reports would correctly extract all information for the past year and the current year up to and including the reporting date's month. However when run from the POS EOD process these programs would not include the data from the current month.

This update corrects this.

<u>Files affected by the enhancement</u>

Source files altered:	SAITMX.CBL	Sales data extract
	SASTRX.CBL	Benchmark data extract
Files to the customer:	sa\SAITMX.GNT	Sales data extract
	sa\SASTRX.GNT	Benchmark data extract
	README.DOC	This file

Date: 3/23/2005
ISS#: ISS221

Changes to Existing Restocking Report

We made the following changed the to Restocking report.

- Add fields item-average-cost, item-price-1, item-status, sales-this year and sales-last-year to the report (not the extract file).
- In order to squeeze the fields in we had to remove the blank character between fields (a blank character will still exist for positive values).
- We split the header into 2 lines for clarity.
- Zero-value fields no longer print a zero character. Zero-value fields now display nothing at all.

Changes to Target Customer Extract

The report has a parameter for indicating the minimum sales amount. The parameter would only allow values greater than zero. We added the ability for the operator to press the F1-key to indicate “All” amounts are to be accepted

Changes to Child Birthday Extract

We added a new report parameter that accepts a range of dates. We also added the Sales History header record to the program. The program now accumulates the sales for the date range entered for each customer processed. It also counts the transactions processed. We added these two new fields to the extract file as well.

We also changed the way children’s names are processed. Previously we were merely adding the customer’s last name to the child’s name to create the child’s full name. Now, if the customer’s name doesn’t require formatting (a company’s name) no change is made to the child’s name at all – it is used as it is entered. If the customer’s name requires formatting the program checks the child’s name to see it is comprised of more than a single part (as in first-name / last-name). If it is it retains the child’s name as-is. If the child’s name is only a single part (presumably the first name) the program concatenates the customer’s last name to the child’s first name to create the child’s full name.

Files affected by the enhancement

Source files altered:	ARCUSX.CBL	New customer birthday extract
	IMRSKP.CBL	Restocking report
	SACUSX.CBL	New target customer extract
Files to the customer:	ar\ARCUSX.GNT	New customer birthday extract
	im\IMRSKP.GNT	Restocking report
	sa\ SACUSX.GNT	New target customer extract
	Readme.doc	This file

Date: 2/8/2005
ISS#: ISS208

Port Prior Modifications to Version 7.5.8

Previously we ported the client's mods to version 7.5.7. However the client intends to stay one year behind in their version of the software so we ported them back to version 7.5.6. At this time we also includes the Corporate update modification created by CSE. However SYN then released a mandatory update for 7.5.8 only that necessitated the client porting their mods to version 7.5.8.

We modified the programs as required to function within the 7.5.8 environment without any changes in the enhancement's functionality or altering newer 7.5.8 functionality.

Note that during this update we renamed the Restocking report from IMRSTP to IMRSKP.

Files affected by the enhancement

Source files altered:	ARCUSX.CBL	New customer birthday extract (2 versions)
	IMBPOP.CBL	Blank PO extract
	IMBSCP.CBL	Best seller by category
	IMCORU.CBL	Corporate update
	IMITMX.CBL	Item data extract
	IMLSSP.CBL	Location stock status
	IMR6WP.CBL	6 Week item history
	IMRSKP.CBL	Restocking report
	IMSBDP .CBL	Item sales by date
	IMYTDP .CBL	Item sales YTD
	PSEODD.CBL	POS EOD driver program
	SACUS2.CBL	Sales by customer report (2 versions)
	SACUSX.CBL	New target customer extract (7.0.150 version)
	SAITMX.CBL	Sales data extract
	SASTRX.CBL	Benchmark data extract
	SYMENU.DAT	Menu definition file
	SQISHL.PL	Text manipulation routines
	SQUEEZ.WS	
	SQUEZR.PL	

Files to the customer:	ar\ARCUSX.GNT	New customer birthday extract (2 versions)
	im\IMBPOP.GNT	Blank PO extract
	im\IMBSCP.GNT	Best seller by category
	im\IMCORU.GNT	Corporate update
	im\IMITMX.GNT	Item data extract
	im\IMLSSP.GNT	Location stock status
	im\IMR6WP.GNT	6 Week item history
	im\IMRSKP.GNT	Restocking report
	im\IMSBDP .GNT	Item sales by date
	im\IMYTDP .GNT	Item sales YTD
	ps\PSEODD.GNT	POS EOD driver program
	sa\ SACUS2.GNT	Sales by customer report
	sa\ SACUSX.GNT	New target customer extract
	sa\SAITMX.GNT	Sales data extract
	sa\SASTRX.GNT	Benchmark data extract
	SYMENU.DAT	Menu definition file
	INSTALL1.BAT	Installation script
	INSTALL2.BAT	Installation script
	README.DOC	This file

Date: 6/17/2004
ISS#: ISS151

Tweaks to Extract Programs in Version 7.5.6 and 7.0.150

Thus update provides a few tweaks to the item extract and the sales extract programs.

Files affected by the enhancement

Source files altered:	IMITMX.CBL	Item data extract
	SAITMX.CBL	Sales data extract
Files to the customer:	im\IMITMX.GNT	Item data extract
	sa\SAITMX.GNT	Sales data extract
	Readme.doc	This file

Date: 6/17/2004
ISS#: ISS151

Port Prior Modifications to Version 7.5.6

Previously we ported the client's mods to version 7.5.7. However the client intends to stay one year behind in their version of the software so we ported them back to version 7.5.6. It also includes the Corporate update modification created by CSE.

We modified the programs as required to function within the 7.5.6 environment without any changes in the enhancement's functionality or altering newer 7.5.6 functionality.

Files affected by the enhancement

Source files altered:	ARCUSX.CBL	New customer birthday extract (2 versions)
	IMBPOP.CBL	Blank PO extract
	IMBSCP.CBL	Best seller by category
	IMCORU.CBL	Corporate update
	IMITMX.CBL	Item data extract
	IMLSSP.CBL	Location stock status
	IMR6WP.CBL	6 Week item history
	IMRSTP.CBL	Restocking report
	IMSBDP .CBL	Item sales by date
	IMYTDP .CBL	Item sales YTD
	PSEODD.CBL	POS EOD driver program
	SACUS2.CBL	Sales by customer report (2 versions)
	SACUSX.CBL	New target customer extract (7.0.150 version)
	SAITMX.CBL	Sales data extract
	SASTRX.CBL	Benchmark data extract
	SYMENU.DAT	Menu definition file
	SQISHL.PL	Text manipulation routines
	SQUEEZ.WS	
	SQUEZR.PL	
Files to the customer:	ar\ ARCUSX.GNT	New customer birthday extract (2 versions)
	im\IMBPOP.GNT	Blank PO extract
	im\IMBSCP.GNT	Best seller by category
	im\IMCORU.GNT	Corporate update
	im\IMITMX.GNT	Item data extract
	im\IMLSSP.INT	Location stock status
	im\IMR6WP.GNT	6 Week item history
	im\IMRSTP.INT	Restocking report
	im\IMSBDP .INT	Item sales by date
	im\IMYTDP .INT	Item sales YTD
	ps\PSEODD.GNT	POS EOD driver program
	sa\ SACUS2.GNT	Sales by customer report
	sa\ SACUSX.GNT	New target customer extract
	sa\SAITMX.GNT	Sales data extract
	sa\SASTRX.GNT	Benchmark data extract
	SYMENU.DAT	Menu definition file
	INSTALL1.BAT	Installation script
	INSTALL2.BAT	Installation script
	README.DOC	This file

Date: 6/6/2004
ISS#: ISS151

Port Prior Modifications to Version 7.5.7

This update ports all of the prior mods from version 7.5.3 to version 7.5.7. It also includes the Corporate update modification created by CSE.

We modified the programs as required to function within the 7.5.7 environment without any changes in the enhancement's functionality or altering newer 7.5.7 functionality.

Files affected by the enhancement

Source files altered:	ARCUSX.CBL	New customer birthday extract (2 versions)
	IMBPOP.CBL	Blank PO extract
	IMBSCP.CBL	Best seller by category
	IMCORU.CBL	Corporate update
	IMITMX.CBL	Item data extract
	IMLSSP.CBL	Location stock status
	IMR6WP.CBL	6 Week item history
	IMRSTP.CBL	Restocking report
	IMSBDP .CBL	Item sales by date
	IMYTDP .CBL	Item sales YTD
	PSEODD.CBL	POS EOD driver program
	SACUS2.CBL	Sales by customer report (2 versions)
	SACUSX.CBL	New target customer extract (7.0.150 version)
	SAITMX.CBL	Sales data extract
	SASTRX.CBL	Benchmark data extract
	SYMENU.DAT	Menu definition file
	SQISHL.PL	Text manipulation routines
	SQUEEZ.WS	
	SQUEZR.PL	

Files to the customer:	ar\ ARCUSX.GNT	New customer birthday extract (2 versions)
	im\IMBPOP.GNT	Blank PO extract
	im\IMBSCP.GNT	Best seller by category
	im\IMCORU.GNT	Corporate update
	im\IMITMX.GNT	Item data extract
	im\IMLSSP.INT	Location stock status
	im\IMR6WP.GNT	6 Week item history
	im\IMRSTP.INT	Restocking report
	im\IMSBDP .INT	Item sales by date
	im\IMYTDP .INT	Item sales YTD
	ps\PSEODD.GNT	POS EOD driver program
	sa\ SACUS2.GNT	Sales by customer report
	sa\ SACUSX.GNT	New target customer extract
	sa\SAITMX.GNT	Sales data extract
	sa\SASTRX.GNT	Benchmark data extract
	SYMENU.DAT	Menu definition file
	INSTALL1.BAT	Installation script
	INSTALL2.BAT	Installation script
	README.DOC	This file

Date: 6/1/2004
ISS#: ISS101 (continued)

Item Data Extract (*Inventory>Utilities>Extract item data*)

This update makes 3 changes to the previously delivered item extract.

- Add 1st received date
- Add prime vendor number
- Replace backslashes in dates with forward slashes

The new extract information record layout is:

Field Name	Format/Size	Type	Comments
YearId	5	Alphanumeric	We are assuming this field holds a 4-digit year
Period	Z9	Numeric	Sales period from SAMOHF.DAT file values 1-13
ItemNumber	15	Alphanumeric	ITEM-NO (left-justified, space-filled if numeric)
CorpStoreNbr	ZZ9	Numeric	From store-# mapping file
ItemDescLine1	25	Alphanumeric	Replace one apostrophe with two apostrophes
AverageCost	ZZZ,ZZ9.999	Numeric	3 decimal places
Price1	ZZZ,ZZ9.999	Numeric	3 decimal places
FirstReceiveDate	mm/dd/yyyy	Alphanumeric	Formatted date field
LastReceiveDate	mm/dd/yyyy	Alphanumeric	Formatted date field
ItemType	1	Alphanumeric	Same as in Counterpoint application
Category	5	Alphanumeric	Same as in Counterpoint application
SubCategory	5	Alphanumeric	Same as in Counterpoint application
UserField1	5	Alphanumeric	Same as in Counterpoint application (class)
UserField2	5	Alphanumeric	Same as in Counterpoint application (subclass)
UserField3	5	Alphanumeric	Same as in Counterpoint application (group)
UserField4	5	Alphanumeric	Same as in Counterpoint application (subgroup)
ItemStatus	1	Alphanumeric	Same as in Counterpoint application
OnHandQty	-ZZZ,ZZZ,ZZ9	Numeric	Integer, prefix with “-“ if negative
CommitQty	-ZZZ,ZZZ,ZZ9	Numeric	Integer, prefix with “-“ if negative
TransferQty	-ZZZ,ZZZ,ZZ9	Numeric	Integer, prefix with “-“ if negative
LayawayQty	-ZZZ,ZZZ,ZZ9	Numeric	Integer, prefix with “-“ if negative
OnOrderQty	-ZZZ,ZZZ,ZZ9	Numeric	Integer, prefix with “-“ if negative
BackOrderQty	-ZZZ,ZZZ,ZZ9	Numeric	Integer, prefix with “-“ if negative
PrimeVendorNumber	12	Alphanumeric	PRIME-VEND-NO (left-justified, space-filled if numeric)

* Each “Z” character is replaced with a space if the character in it is a zero.
* Each “9” character will always contain a digit of “0” through “9”.
* Once the fields are loaded, all embedded spaces are removed as the information is left justified.

Files affected by the enhancement

Source files altered:	IMITMX.CBL	Item data extract
Files to the customer:	im\IMITMX.GNT	Item data extract
	README.DOC	This file

Date: 11/13/2003
ISS#: ISS101

Create Automated Data Extract System

The client has a corporate office and several stores that are not connected via a WAN or MultiSite. They wanted the individual stores to automatically create 3 sets of extract files on a monthly basis that would then be transmitted to the corporate office using the client's own procedures.

We wrote three new programs (along with a couple more changes) to create the desired extract files.

POS End of Day Posting Program

The client wanted to automate the running of these programs so they would run once a month. We added the code required to call the extract programs from the POS EOD posting driver program. This program is run each day. We changed the EOD posting driver program to run the extract programs after the regular posting process has completed whenever the system date is the same as the current period's end date. Thus the extract programs are automatically run each time the POS EOD process is a run on the last day of each period without having to setup any processes outside of CounterPoint. (Actually, without having to set up anything at all.) The programs run without prompting for a printer (they default to "Print to disk") or for parameters.

CounterPoint Menu Definitions

We added 3 separate menu selections from which to run the extract programs in the event one or all of them have to be run manually. The three extract programs behave slightly differently when run from the menus. When invoked from the CounterPoint menus all three programs prompt the operator for the print device. When the programs are called during the posting process they all default to "Print to disk" instead. This allows the posting process to run without attendance and provides an operator an option when running the extracts manually. Additionally, when run from the CounterPoint menu, the sales data extract and the benchmark data extract programs prompt the operator for a single parameter, the report date. This date indicates to the extract program through which month the extracted information should be retrieved. If the date entered is the last date of the period the information for that month (and the preceding months) will be extracted. If the date is not the last date of the period, only the information for the prior months is extracted since the date entered does not contain an entire month. The item data extract only extracts the current information for the items – not the information as of the indicated date so it does not prompt for a date parameter.

Regional Store-# Mapping File

Unfortunately, many of the stores have been setup to use the same store numbers. Obviously we could not allow several stores to transmit their data to the corporate office using the same store number. To resolve this, the client will create an ASCII file at each store's location that translates that location's store numbers to the store numbers to be reported to the corporate office. This way the data reported to the corporate office from the various locations will all have a unique store number. This file must be named STOREMAP.TXT and must reside in each site's Counterpoint top-level directory. It must contain the mappings for one or more stores. (We are assuming that no sites have multiple companies defined using the same store numbers.) Blank lines or lines with a semi-colon character (";") in column 1 will be ignored by the extract programs. This will allow the client the ability to add comments to the file. If the file is missing, does not contain an entry for the store being processed or does not contain a valid value for the replacement store number, the extract program will abort or partially process the information for the site and write an appropriate message to an exception report.

This store-# mapping file will be an ASCII file with the format of:

Field Name	Length	Type	Comments
LocalStoreNbr	3	Numeric	This is the system's store number in CounterPoint
CorpStoreNbr	3	Numeric	This is the store-# to be assigned in the extract files
CorpStoreName	40	Alphanumeric	Replacement store name. (Commentary)

Regional Location Mapping File

Similar to the store-# mapping situation above, we have the same situation with locations. The Sales Data Extract program does not deal with stores because inventory is not managed by stores. It manages inventory by locations. Generally speaking, most CounterPoint users set their location codes to mimic their store numbers, but this is not always true. Each store can have its own default location code. In fact several stores can use the same default location code.

To handle this we have defined another ASCII file at each store's location to translate that installation's locations to the location to be reported to the corporate office. This way the data reported to the corporate office from the various locations will all have a unique location values. This file must be named LOCMAP.TXT and must reside in each site's Counterpoint top-level directory. It must contain the mappings for one or more locations. (We are assuming that no sites have multiple companies defined using the same location code.) Blank lines or lines with a semi-colon character (";") in column 1 will be ignored by the item data extract program. This will allow the client the ability to add comments to the file. If the file is missing, does not contain an entry for the location being processed or does not contain a valid value for the replacement location code, the extract program will abort or partially process the information for the site and write an appropriate message to an exception report.

This store-# mapping file will be an ASCII file with the format of:

Field Name	Length	Type	Comments
LocalLocationCode	4	Alphanumeric	This is the system's location code in CounterPoint
CorpLocationCode	4	Alphanumeric	This is the location code to be assigned in the extract files
CorpStoreName	40	Alphanumeric	Replacement store name. (Commentary)

Item Data Extract (*Inventory>Utilities>Extract item data*)

The client wanted the ability to extract current item inventory levels for the items at each of its store locations. These records would represent a snap-shot of the store’s items at the time the extract is run. Thus there is only a single record written for each item at each site.

The extracted information is written to a pipeline-delimited ASCII file with the name ITEMS.TXT containing the following fields:

Field Name	Format/Size	Type	Comments
YearId	5	Alphanumeric	We are assuming this field holds a 4-digit year
Period	Z9	Numeric	Sales period from SAMOHF.DAT file values 1-13
ItemNumber	15	Alphanumeric	ITEM-NO (left-justified, space-filled if numeric)
CorpStoreNbr	ZZ9	Numeric	From store-# mapping file
ItemDescLine1	25	Alphanumeric	Replace one apostrophe with two apostrophes
AverageCost	ZZZ,ZZ9.999	Numeric	3 decimal places
Price1	ZZZ,ZZ9.999	Numeric	3 decimal places
LastReceiveDate	mm/dd/yyyy	Alphanumeric	Formatted date field
ItemType	1	Alphanumeric	Same as in Counterpoint application
Category	5	Alphanumeric	Same as in Counterpoint application
SubCategory	5	Alphanumeric	Same as in Counterpoint application
UserField1	5	Alphanumeric	Same as in Counterpoint application (class)
UserField2	5	Alphanumeric	Same as in Counterpoint application (subclass)
UserField3	5	Alphanumeric	Same as in Counterpoint application (group)
UserField4	5	Alphanumeric	Same as in Counterpoint application (subgroup)
ItemStatus	1	Alphanumeric	Same as in Counterpoint application
OnHandQty	-ZZZ,ZZZ,ZZ9	Numeric	Integer, prefix with “-“ if negative
CommitQty	-ZZZ,ZZZ,ZZ9	Numeric	Integer, prefix with “-“ if negative
TransferQty	-ZZZ,ZZZ,ZZ9	Numeric	Integer, prefix with “-“ if negative
LayawayQty	-ZZZ,ZZZ,ZZ9	Numeric	Integer, prefix with “-“ if negative
OnOrderQty	-ZZZ,ZZZ,ZZ9	Numeric	Integer, prefix with “-“ if negative
BackOrderQty	-ZZZ,ZZZ,ZZ9	Numeric	Integer, prefix with “-“ if negative

* Each “Z” character is replaced with a space if the character in it is a zero.
* Each “9” character will always contain a digit of “0” through “9”.
* Once the fields are loaded, all embedded spaces are removed as the information is left justified.

Most of the information for this extract comes from the item file, which contains the information for all locations. The program reads any inventory records for the item and load the most recent date into the field LastReceiveDate. The YearId and Period fields will represent the system’s current month in the current calendar.

Note that the item file has no reference to the store number (or location) in its index. So the item information being extracted is for ALL locations and stores within the company. So this program does not map the current store number to a corporate store number (or location codes). It merely uses the first corporate store number it finds in the STOREMAP.TXT file as the store number in the extract file.

<u>Sales Data Extract *(Sales history>Utilities>Extract sales data)*</u>

The client wanted the ability to extract sales data from each store grouped by months. The monthly history file (SAMOHF.DAT) has an ideal record type to use as a source for this data. This record type contains the sales data for the activity associated with the items within the company (for all stores and locations). However, the monthly history file is not keyed by the numerical year, but by the year-id as setup in the calendar file. Nor is it set up to automatically store the months in calendar months. The values are accumulated by the ‘months’ as they are defined in the calendar file. Thus the monthly history file may accumulate the sales information by calendar month or 4-week months or even by quarters of the year. The point is that the ‘months’ that are reported by the monthly history file are defined by the ‘months’ that are defined within the company’s calendar file. Which may also vary from installation to installation. In order for the comparison of the resulting data to make sense (in an apples-to-apples manner), all sites will have to maintain the same calendar months.

The program retrieves the current year-id from *Setup>System>Company>8. Current year ID*. It reads that calendar record to obtain the year-id for the prior year. The extract program then creates a set of item records for each item on file. Each set is comprised of the monthly sales history for the prior year (up to 13 records depending on how the year was defined for that calendar year) and a record for each month of the current year (as defined on the calendar record). The system’s current month is used as the cut-off for processing the current year’s information. If a monthly history record does not exist for the prior year or the current year, that set of records is not written. So each set of records for an item may be comprised of between 0 and 26 records inclusively.

The extracted information is written to a pipeline-delimited ASCII file with the name SALES.TXT containing the following fields:

<u>Field Name</u>	<u>Format/Size</u>	<u>Type</u>	<u>Comments</u>
CorpLocation	4	Alphanumeric	From location-code mapping file
ItemNumber	15	Alphanumeric	ITEM-NO (left-justified, space-filled if numeric)
YearId	5	Alphanumeric	We are assuming this field holds a 4-digit year
Period	Z9	Numeric	Sales period from SAMOHF.DAT file values 1-13
QuantitySold	-Z,ZZZ,ZZZ,ZZ9	Numeric	Integer, prefixed with “-“ if negative
QuantityReturned	-Z,ZZZ,ZZZ,ZZ9	Numeric	Integer, prefixed with “-“ if negative
SalesAmount	-ZZ,ZZZ,ZZ9.99	Numeric	2 decimal places, prefixed with “-“ if negative
CostAmount	-ZZ,ZZZ,ZZ9.99	Numeric	2 decimal places, prefixed with “-“ if negative

* Each “Z” character is replaced with a space if the character in it is a zero.
* Each “9” character will always contain a digit of “0” through “9”.
* Once the fields are loaded, all embedded spaces are removed as the information is left justified.

Fields QuantitySold and QuantityReturned will generally both be positive values.

The monthly history records this program uses have a reference to the location within the index. The client wanted the extracted records to be extracted by the store number, but as noted at the beginning of this document, we cannot use the original store numbers. (In fact we cannot use store numbers at all.) This extract program will load the first 50 records from the LOCMAP.TXT file to translate the location codes encountered to the location codes to be reported in the extract file.

Benchmark Data Extract *(Sales history>Utilities>Extract benchmark data)*

The client wanted an extract file to be created at each store containing basic ticket statistics for that site. The monthly history file has another record type containing this information. It carries the total sales, total cost, #-of tickets and #-of-lines for the store broken down by the same 'month' periods as described by the sales extract section of this document. Using these fields directly or by calculations thereof we obtained the desired information.

As in the Sales extract program described above, this extract creates a set of records. This extract creates a set of up 26 records for each store containing the information for the prior year and the months to date in the current year. See the description of how the sales extract program handles the calendar file for an overview of how the periods are to be reported.

The extracted information is written to a pipeline-delimited ASCII file with the name STATS.TXT containing the following fields:

Field Name	Format/Size	Type	Comments
CorpStoreNbr	ZZ9	Numeric	From store-# mapping file
YearId	5	Alphanumeric	We are assuming this field holds a 4-digit year
Period	Z9	Numeric	Sales period from SAMOHF.DAT file values 1-13
TicketsPerMonth	ZZZ,ZZ9	Numeric	Integer
LinesPerMonth	Z,ZZZ,ZZ9	Numeric	Integer
AvgLinesPerTicket	Z9.99	Numeric	2 decimal places
AvgSalePerLine	-ZZ,ZZ9.99	Numeric	2 decimal places, prefixed with "-" if negative
AvgTicketValue	-ZZZ,ZZ9.99	Numeric	2 decimal places, prefixed with "-" if negative
AvgGPP	-ZZ9.99	Numeric	2 decimal places, prefixed with "-" if negative

* Each "Z" character is replaced with a space if the character in it is a zero.
* Each "9" character will always contain a digit of "0" through "9".
* Once the fields are loaded, all embedded spaces are removed as the information is left justified.

TicketsPerMonth and LinesPerMonth will be reported directly from the monthly history record. The averaged fields will be calculated.

The monthly history records this program uses have a reference to the store number within the index. The client wanted the extracted records to be extracted by the store number, but as noted at the beginning of this document, we could not use the original store numbers. This extract program loads the first 50 records from the STOREMAP.TXT file to translate the store numbers encountered to the store numbers to be reported on the extract file.

<u>Exception Report</u>

Each extract program creates an exception report to which they write error conditions. Depending on the extract being run the programs reports different errors.

<u>Sales extract program</u>

<u>Error</u>	<u>Action taken</u>
• STOREMAP.TXT file is missing.	Extract not run
• Local store-# not found in the STOREMAP.TXT file.	
• Item has no sales history for prior year.	No prior year info extracted for item
• Item has no sales history for the current year.	No current year info extracted for item
• Item has no sales history for this year or last year.	No info extracted for item

<u>Item extract program</u>

<u>Error</u>	<u>Action taken</u>
• STOREMAP.TXT file is missing.	Extract not run
• Item has no inventory records (thus no last received info).	Item extracted sans the missing field.

<u>Benchmark extract program</u>

<u>Error</u>	<u>Action taken</u>
• STOREMAP.TXT file is missing.	Extract not run
• Local store-# not found in the STOREMAP.TXT file.	Info for that store number not extracted
• Store has no sales history for prior year.	No prior year info extracted for store
• Store has no sales history for the current year.	No current year info extracted for store
• Store has no sales history for this year or last year.	No info extracted for store

<u>Installation instructions</u>

The modified programs were emailed to the client loose. We also updated the installation sets and emailed them to the client. The installation instructions for the set are the same as in the initial delivery of the mods.

<u>Files affected by the enhancement</u>

Source files altered:	IMITMX.CBL	Item data extract
	PSEODD.CBL	POS EOD driver program
	SAITMX.CBL	Sales data extract
	SASTRX.CBL	Benchmark data extract
	SYMENU.DAT	Menu definition file
	SQUEEZ.WS	Text manipulation routines
	SQUEZR.PL	
	SQISHL.PL	
Files to the customer:	im\IMITMX.GNT	Item data extract
	ps\PSEODD.GNT	POS EOD driver program
	sa\SAITMX.GNT	Sales data extract
	sa\SASTRX.GNT	Benchmark data extract
	SYMENU.DAT	Menu definition file
	INSTALL.BAT	Installation script
	README.DOC	This file

Date: 5/29/2003
ISS#: ISS056

Change Delimiters in Extract Programs

The client currently uses several report/export programs (created by Synchronics) to create a Comma Separated Value (CSV) files for further processing of item sales data. However, using the comma character as a delimiter has proved to be problematic since some of the data fields contain the comma character. This results in fields being parsed incorrectly when the file is imported.

We changed all 7 extract programs to create the export file in the same format as it is currently but using the pipeline character (“|”) as a delimiter. This character is rarely (if ever) used in data entry and (once the importing program has been changed to match) will allow the file to be imported without field parsing errors.

The source code provided us was written in version 7.0.150 and 7.3.2. We made the changes to the 7.0.150 programs as-is (as-are?). We ported the 7.3.2 programs to version 7.5.3 for a total of 14 delivered programs.

Installation instructions

The modified programs were emailed to the client loose. We also updated the installation sets and emailed them to the client. The installation instructions for the set are the same as in the initial delivery of the mods.

Files affected by the enhancement

Source files altered:	IMBPOP.CBL	Blank PO extract
	IMBSCP.CBL	Best seller by category
	IMLSSP.CBL	Location stock status
	IMR6WP.CBL	6 Week item history
	IMRSTP.CBL	Restocking report
	IMSBDP .CBL	Item sales by date
	IMYTDP .CBL	Item sales YTD

Files to the customer:	im\IMBPOP.GNT	Blank PO extract	
	im\IMBSCP.GNT	Best seller by category	
	im\IMLSSP.INT	Location stock status	(7.5.3 version is a GNT)
	im\IMR6WP.GNT	6 Week item history	
	im\IMRSTP.INT	Restocking report	(7.5.3 version is a GNT)
	im\IMSBDP .INT	Item sales by date	(7.5.3 version is a GNT)
	im\IMYTDP .INT	Item sales YTD	(7.5.3 version is a GNT)
	INSTALL.BAT	Installation script	
	README.DOC	This file	

Date: 4/1/2003
ISS#: ISS033 (Continued)

Create Target Customer Extract Program

We ported the 7.5.3 version of this report to 7.0.150 and delivered it.

New Customer Birthday Extract Program

The client wanted us to quote what it would cost to split the customer’s first and last names apart on the extracted file. Since it was a relatively minor change we provided this tweak for free since the client has been so patient in waiting for the Target customer Extract program getting ported to 7.0.150.

Rather than replace the customer name in the extract file we added the customer’s first and last name to the end of the file only if the name has been tagged as a person’s name n CounterPoint. If the name is a business it will not appear in the new fields. Either way we are still formatting and extracting the customer’s name in the first field of the extract file.

Installation instructions

The modified programs were emailed to the client loose. We also updated the installation sets and emailed them to the client. The installation instructions for the set are the same as in the initial delivery of the mods.

Files affected by the enhancement

Source files altered:	ARCUSX.CBL	New customer birthday extract (2 versions)
	SACUSX.CBL	New target customer extract (7.0.150 version)
Files to the customer:	ar\ ARCUSX.GNT	New customer birthday extract (2 versions)
	sa\ SACUSX.GNT	New target customer extract (7.0.150 version)
	INSTALL.BAT	Installation script (2 versions)
	README.DOC	This file

Date: 3/4/2003
ISS#: ISS038

This enhancement actually updated the next enhancement as it was being produced. The client wanted us to add the number of tickets for each customer to the Sales by customer report and the new Target customer extract program. Since we had only just begun to create the Target customer extract program we included this request in the new program during its initial creation. Thus we will not list it in this section. However the Sales by Customer program had already been delivered so it's modifications are shown here.

<u>Files affected by the enhancement</u>

Source files altered:	SACUS2.CBL	Sales by customer report (2 versions)
Files to the customer:	sa\ SACUS2.GNT	Sales by customer report (7.5.3 version)
	sa\ SACUS2.INT	Sales by customer report (7.0.150 version)
	INSTALL.BAT	Installation script (2 versions)
	README.DOC	This file

Date: 2/11/2003
ISS#: ISS033

This enhancement is actually a set of 3 enhancements. The first is a tweak to an existing modification. The others are new extract programs. Each is detailed below.

Installation instructions
The modified programs were emailed to the client loose as they were completed. Once all programs were tested we updated the installation script, repackaged the enhancements and emailed the updated installation sets to the client. The installation instructions for the set are the same as in the initial delivery of the mods.

Mod to Export Customer Info in Sales Analysis by Customer Report

In a prior mod we added the ability to export basic customer info in the Sales Analysis by Customer report. This enhancement adds the customer's email address to the exported file.

Files affected by the enhancement

Source files altered:	SACUS2.CBL	Sales by customer report (2 versions)
Files to the customer:	sa\ SACUS2.GNT	Sales by customer report (7.5.3 version)
	sa\ SACUS2.INT	Sales by customer report (7.0.150 version)
	INSTALL.BAT	Installation script (2 versions)
	README.DOC	This file

New Customer Birthday Extract Program

The client wanted the ability to create an ASCII file containing basic customer information, customer profile information and sales data delimited by the pipeline character ("|").

We created a new program that extracts the information from the sales history file and writes it to an ASCII file named CHLDBDAY.TXT in the CounterPoint top-level directory. If the file already exists it is overwritten with the new file. The extract file will contain the following fields in this order:

- Customer number
- Customer name (formatted first-name, space, last name when appropriate)
- Address-1
- Address-2
- City
- State
- Zip
- Email address (See below)
- Last sale date (formatted as a string such as mm/dd/yy)
- Last sale amount (formatted -ZZZ,ZZZ,ZZ9.99 where a zero in a 'Z' position is replaced with a space and a zero in a '9' position is replaced by the '0' character.
- Child name (as entered in customer profile field)
- Child birthdate (formatted as a string such as mm/dd/yy)

The fields are variable in length with a pipeline character delimiting them.

The customer info fields come directly from the customer record. Note the email address field does not exist in version 7.0.150. The client will enter the customer's email address in the contact 1 & 2 fields. The 7.0.150 version of the extract program concatenates these two fields together to create the email address. Version 7.5.3 already has the customer's email address field from which we extract this field. The last sale date and amount fields will come directly from the customer record. The child name and birthdate fields come from the customer's profile fields. CounterPoint allows the definition of up to 15 profile fields which may be assigned a unique value for each customer. The client will set these profile fields to retain the name and birthdate of the customer's children (up to 7). The profile fields are assigned to each child in pairs the first being the child's name and the second the child's birthdate.

The new parameter screen resembles:

```
Customer birthday extract



        Please enter:

            1. Birth month        99xxxxxxX

            2. Cutoff birth year  9999




Field number to change ?
```

`1. Birth month`	Allow a value of 1 through 12. Display month to the right.
`2. Cutoff birth year`	Allow a 4-digit value between 1900 and the current year (inclusively).

Once the parameters are entered and the ENTER key pressed at "Field number to change ?" the program runs to completion. The data is exported in customer number sequence. No report is generated.

The program reads through the customer file (by customer #) examining the profile fields. If there are no children defined in the profile fields the customer is skipped and the next is retrieved. If children have been defined in the profile fields the program examines each child's birthdate to determine if the date passes the parameter filters. If the child's birth-month matches the birth-month-parameter and the child's birthdate-year is less than or equal to the cutoff-birth-year parameter, an extract record is written for the customer with that child's name and birthdate. If not, the next child's (for that customer) profile information is examined in turn until a valid child's information is encountered or there is no more profile information for the customer. If more than one child for a customer passes the parameter filters, then one extract record is written for each child. Thus a single customer record could, in theory, create up to 7 extract records (each for a different child).

Note that this program was created for versions 7.0.150 and 7.5.3. These two versions require a different compiler be used when creating the programs so the two programs should not be mixed up. Because of the difference in the handling of the customer email address field, the programs are also functionally different.

<u>Files affected by the enhancement</u>

Source files altered:	ARCUSX.CBL	New customer birthday extract (2 versions)
	SQUEEZ.WS	Text manipulation routines
	SQUEZR.PL	
	SQISHL.PL	
Files to the customer:	ar\ ARCUSX.CBL	New customer birthday extract (2 versions)
	INSTALL.BAT	Installation script (2 versions)
	README.DOC	This file

Create Target Customer Extract Program

The client wanted the ability to create an ASCII file containing basic customer information and sales data delimited by the pipeline character ("|").

We created a new program that extracts the information from the sales history file and writes it to an ASCII file named TRGTCUST.TXT in the CounterPoint top-level directory. If the file already exists it is overwritten with the new file. The extract file contains the following fields in this order:

- Customer number
- Customer name (formatted first-name, space, last name when appropriate)
- Address-1
- Address-2
- City
- State
- Zip
- Email address (See below)
- First sale date (formatted as a string such as mm/dd/yy)
- First sale amount (formatted -ZZZ,ZZZ,ZZ9.99 where a zero in a 'Z' position is replaced with a space and a zero in a '9' position is replaced by the '0' character.
- Last sale date (formatted as a string such as mm/dd/yy)
- Last sale amount (formatted -ZZZ,ZZZ,ZZ9.99 where a zero in a 'Z' position is replaced with a space and a zero in a '9' position is replaced by the '0' character.
- Total sales amount (formatted -ZZZ,ZZZ,ZZ9.99 where a zero in a 'Z' position is replaced with a space and a zero in a '9' position is replaced by the '0' character.

The fields are variable in length with a pipeline character delimiting them.

The customer info fields come directly from the customer record. Note the email address field does not exist in version 7.0.150. The client will enter the customer's email address in the contact 1 & 2 fields. The 7.0.150 version of the extract program concatenates these two fields together to create the email address. Version 7.5.3 already has the customer's email address field from which we extract this field. The first and last sale date and amount fields are extracted from the sales history file.

The new parameter screen resembles:

```
Target customer extract

   Please enter:

      1. Customer range       XXXXXXXXXXXX to XXXXXXXXXXXX
      2. Customer category    XXXXX XXXXX XXXXX XXXXX XXXXX XXXXX XXXXX XXXXX
      3. Customer zip code    XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX >>

      4. Item #               XXXXXXXXXXXXXXX XXXXXXXXXXXXXXX >>
      5. Item vendor #        XXXXXXXXXXXX XXXXXXXXXXXX XXXXXXXXXXXX >>
      6. Item category        XXXXX XXXXX XXXXX XXXXX XXXXX XXXXX XXXXX XXXXX
      7. Item sub-category    XXXXX XXXXX XXXXX XXXXX XXXXX XXXXX XXXXX XXXXX
      8. Item user1           XXXXX XXXXX XXXXX XXXXX XXXXX XXXXX XXXXX XXXXX
      9. Item user2           XXXXX XXXXX XXXXX XXXXX XXXXX XXXXX XXXXX XXXXX

     10. Store #              999xx99 999 999 999 999 999 999

     11. Date range           99/99/99xx to 99/99/99

     12. Minimum sale amount  999,999,999


Field number to change ?  ___
```

1. Customer range	Allow F1 keys for “First” to “Last”. Allow F9 data lookup. Start-cust-# must be <= End-cust-#. If Start-cust-# = End-cust-#, cust-# must be on file.
2. Customer category	Allow F1 key for “All” for 1st entry. Allow F9 data lookup.
3. Customer zip code	Allow Allow F1 key for “All”. Allow F2 key to enter up to 12 values in a popup window.
4. Item #	Allow F1 key for “All”. Allow F2 key to enter up to 8 values in a popup window. Allow F9 data lookup here and in popup window.
5. Item vendor #	Allow Allow F1 key for “All”. Allow F2 key to enter up to 8 values in a popup window. Allow F9 data lookup here and in popup window.
6. Item category	Allow F1 key for “All” for 1st entry. Allow F9 data lookup.
7. Item sub-category	Allow F1 key for “All” for 1st entry. Allow F9 data lookup.
8. Item *user1*	Allow F1 key for “All” for 1st entry. Allow F9 data lookup.
9. Item *user2*	Allow F1 key for “All” for 1st entry. Allow F9 data lookup.
10. Store #	Allow F1 key for “All” for 1st entry. Allow F9 data lookup.
11. Date range	Allow F1 keys for “Earliest” to “Latest”. Start-date must be <= End-date. End-date <= today.
12. Minimum sale amount	Must be >= zero.

*** Note** In the sections above replace the littorals *user1* and *user2* with the values provided these tags in Setup…Inventory…Control.

Once the parameters are entered and the ENTER key pressed at “Field number to change ?” the program runs to completion. The data is exported in customer number sequence. No report is generated.

The program reads through the customer file (by customer #) filtering out customers that fail to match the edits for the customer-category and zip-code parameters. Once a valid customer has been found the program attempts to retrieve valid sales history records for the customer. First it attempts to find valid ticket header records using the date range and store-# parameters as filters. If the header passes these filters the line items for the header are read in the order in which they were entered. The current item record for each line item is retrieved. If the item is an inventory-type item it will then be compared to the item parameters (item-#, vendor-#, category, sub-category, *user1* and *user2* fields). If the line item passes these filters the ticket discount percentage (if any) is applied to the line item's extended price and the resulting amount accumulated for the customer. Once all of the valid tickets for the customer have been read and processed, the accumulated amount of sales are compared to the minimum-sale-amount parameter. If the accumulated sales amount is greater then or equal to the parameter, an extract record is created for the customer. The accumulated sales amount is written to the extract file.

Note that this program was created for version 7.0.150 and 7.5.3. These two versions require a different compiler be used when creating the programs so the two programs should not be mixed up. Because of the difference in the handling of the customer email address field, the programs are also functionally different.

<u>Included with this proposal</u>

- Create the new extract program as described above for 2 different versions of CounterPoint.
- Add the menu selection to both versions of the CounterPoint graphical menus.
- A simple installation procedure.
- Documentation of the enhancement and the files affected.
- A 30-day guarantee that the software operates as designed in this proposal.

<u>Files affected by the enhancement</u>

Source files altered:	SACUSX.CBL	New target customer extract (2 versions)
	SQUEEZ.WS	Text manipulation routines
	SQUEZR.PL	
	SQISHL.PL	
Files to the customer:	sa\ SACUSX.CBL	New target customer extract (2 versions)
	INSTALL.BAT	Installation script (2 versions)
	README.DOC	This file

Date: 10/7/2002
ISS#: ISS018

Export Customer Info in Sales Analysis by Customer Report

The client needed the ability to create an ASCII file containing basic customer information delimited by the pipeline character (“|”). The CounterPoint ‘Sales Analysis by Customer’ report had the ideal parameters for the utility and made an excellent skeleton for the new program.

We created a new report (“Export by customer”) based on the existing Sales Analysis by Customer report that exports the data reported into an ASCII file named CSTBYSLS.TXT. The file is created in the CounterPoint top-level directory and contains the following fields in this order:

- **Customer number**
- **Customer name** (formatted using standard CounterPoint routines as appropriate)
- **Address-1**
- **Address-2**
- **City**
- **State**
- **Zip**
- **Last sale date** (formatted as a string such as mm/dd/yy)
- **Last sale amount** (formatted -ZZZ,ZZZ,ZZ9.99 where a zero in a ‘Z’ position is replaced with a space and a zero in a ‘9’ position is replaced by the ‘0’ character.
- **Reported sales total** (formatted -ZZZ,ZZZ,ZZ9.99 where a zero in a ‘Z’ position is replaced with a space and a zero in a ‘9’ position is replaced by the ‘0’ character.

The fields are variable in length with a pipeline character delimiting them.

The report parameters are the same as the existing report's except parameters 13 and 14 (the additional reporting periods) are omitted. A new parameter '13 Create export file ?' has been added to control the generation of the export file.

The new report parameter screen resembles:

```
Sales analysis by customer

       Please enter:

            1. Print in order by             Xxxxxxxxxxxxxxxxxxx
            2. Customer range                XXXXXXXXXXX to XXXXXXXXXXX
            3. Customer type                 Xxxxxxxxxx
            4. Customer category range       "First" to "Last"
            5. Sales rep range               XXXXXXXXXX to XXXXXXXXXX
            6. State range                   "First" to "Last"
            7. Zip code range                XXXXXXXXXX to XXXXXXXXXX

            8. Print detail ?                X   1 page per xxxxxxxx ? X
            9. Dollar rounding ?             X
           10. Number of customers to print  999,999
           11. Cutoff sales amount           999,999,999
               Print custs with no sales ?   Y

           12. Report period                 Xxxxxxxxxxxxxxxxxxxxxxxxxxx

           13. Create export file ?          X


Field number to change ?  ___
```

The data is exported in the order specified by parameter '1. Print in order by'. The export file is only created if the report is run in detail format.

If the export file already exists it is overwritten.

Note that the new program still functions as a report. It merely reports 2 less reporting periods and has the ability to export the customer information.

We added this new report to the CounterPoint menus at 'Sales history…Reports…Sales analysis reports…Export by customer'. No other menu selections were altered and no file layouts affected.

Note that this program was created for versions 7.0.150 and 7.5.3. These two versions have different source code bases and require a different compiler when creating the programs so the two programs should not be mixed up.

<u>Included with this proposal</u>

- Create the new report as described above for 2 different versions of CounterPoint.
- Documentation of the enhancement and the files affected.
- Two simple installation procedures.
- A 30-day guarantee that the software operates as designed in this proposal.

Installation instructions

Please note that there are two separate installation diskettes associated with this enhancement. There is a version 7.5.3 diskette and a version 7.0.150 diskette. The diskettes have the same filenames within them but each has its own version-specific copies of the menu definition file, the new program and the installation script. The Readme.doc is the same on both disks. The installation instructions are the same for both versions. However installing the wrong diskette at a site will result in messed-up menus and a mod that will not work.

To perform the installations:

1. As a precaution, perform a complete backup of your entire CounterPoint system (data and programs).

2. This procedure must be run from the DOS command line. If you are running Microsoft Windows, open an MS-DOS window to work in or reboot your computer into MS-DOS mode.

3. Have all users exit the CounterPoint system completely.

4. Place the update diskette into your diskette drive. Change to the diskette drive by typing:

   `A:` <ENTER> (If the diskette is drive A:)

5. Invoke the installation script by typing the following command:

   `INSTALL C:\SYN` <ENTER>

   Replace "C:\SYN" with the drive:\directory in which CounterPoint has been installed. Note that if you are not performing this installation on the PC that has CounterPoint loaded on it (you are performing the installation over a network), you will need to map a drive to the CounterPoint machine to use this syntax. The modified programs are installed into their respective directories. The original programs are saved by renaming the extensions to a number that represents the date the enhancement was created.

6. When the installation has completed, change back to the logical drive in which CounterPoint has been installed by typing:

   `C:` <ENTER>

7. Remove the diskette and store it in a safe place.

This completes the installation procedure. Your CounterPoint system will now have been updated with the enhancements as per the specifications. Should you require assistance, please feel free to contact our office.

Files affected by the enhancement

Source files altered:	SACUS2.CBL	New S/A Export by customer report (2 versions)
	SYMENU.DAT	Menu definition file (2 versions)
	SQUEEZ.WS SQUEZR.PL SQISHR.PL	Text manipulation routines
Files to the customer:	sa\SACUS2.GNT	New S/A Export by customer report (2 versions)
	SYMENU.DAT	Menu definition file (2 versions)
	INSTALL.BAT	Installation script (2 versions)
	README.DOC	This file

```
Date 99/99/99  Time 16:49:41                             DEVENS, MA 01432                         Report# 99999  User Z    Page 0001

                                           D A I L Y   R E S T O C K I N G   R E P O R T

    Vendor-#: XXXXXXXXXXXX Xxxxxxxxxxxxxxxxxxxxxxxx B Y   V E N D O R / I T E M
Categ/SubCat: "All"/"All"
  Time-range: 99:99xxxxx to 99:99xxx

------------------------------------------------------------------------------------------------------------------------------------
Item-#           Description-1             Status    Sold    Avail     Xfer    OnOrd   QtyYTD    Vendor        Categ/SubCat
------------------------------------------------------------------------------------------------------------------------------------

XXXXXXXXXXXXXXX  Xxxxxxxxxxxxxxxxxxxxxxxxx    X     ZZZZZ-   ZZZZZ-   ZZZZZ-   ZZZZZ-   ZZZZZ-   XXXXXXXXXXXX  XXXXX/XXXXX

XXXXXXXXXXXXXXX  Xxxxxxxxxxxxxxxxxxxxxxxxx    X     ZZZZZ-   ZZZZZ-   ZZZZZ-   ZZZZZ-   ZZZZZ-   XXXXXXXXXXXX  XXXXX/XXXXX

XXXXXXXXXXXXXXX  Xxxxxxxxxxxxxxxxxxxxxxxxx    X     ZZZZZ-   ZZZZZ-   ZZZZZ-   ZZZZZ-   ZZZZZ-   XXXXXXXXXXXX  XXXXX/XXXXX

XXXXXXXXXXXXXXX  Xxxxxxxxxxxxxxxxxxxxxxxxx    X     ZZZZZ-   ZZZZZ-   ZZZZZ-   ZZZZZ-   ZZZZZ-   XXXXXXXXXXXX  XXXXX/XXXXX

XXXXXXXXXXXXXXX  Xxxxxxxxxxxxxxxxxxxxxxxxx    X     ZZZZZ-   ZZZZZ-   ZZZZZ-   ZZZZZ-   ZZZZZ-   XXXXXXXXXXXX  XXXXX/XXXXX

XXXXXXXXXXXXXXX  Xxxxxxxxxxxxxxxxxxxxxxxxx    X     ZZZZZ-   ZZZZZ-   ZZZZZ-   ZZZZZ-   ZZZZZ-   XXXXXXXXXXXX  XXXXX/XXXXX


Total Items Sold: ZZZ,ZZ9
  Total Qty Sold: ZZZ,ZZ9
```

```
Date 99/99/99  Time 99:99:99                          Camptown Hardware, Inc.                      Report# 9999  User XXX  Page 9999

                                M U L T I - L O C A T I O N   S T O C K   S T A T U S   R E P O R T

 Item type: Xxxxxxxxxxxxxxxx              XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX    Sales range: ZZ,ZZZ,ZZ9- to ZZ,ZZZ,ZZ9-
Item range: XXXXXXXXXXXXXXX to XXXXXXXXXXXXXXX                                       Qty sold date range: 99/99/99xx to 99/99/99
 Locations: XXXXX XXXXX XXXXX XXXXX                                                       Qty sold range: ZZ,ZZZ,ZZ9- to ZZ,ZZZ,ZZ9-
    Vendor: XXXXXXXXXXXX Xxxxxxxxxxxxxxxxxxxxxxxxx                                    Qty on order range: ZZ,ZZZ,ZZ9- to ZZ,ZZZ,ZZ9-
  Category/sub-category: XXXXXxx / XXXXXxx                                             Qty on hand range: ZZ,ZZZ,ZZ9- to ZZ,ZZZ,ZZ9-
user1/user2/user3/user4: XXXXX/XXXXX/XXXXX/XXXXX                                  Corporate file created:  Store-#: 999
                                                                                      Local file created: Filename: Xxxxxxxxxxxx

------------------------------------------------------------------------------------------------------------------------------------
                                                                  |-----Loc:XXXX------| |-----Loc:XXXX------| |-----Grp:XXXXX-----|
Item-#          Description-1             Cost  Price Sub Cls Pak  Sales  Qty   OH   OO  Sales  Qty   OH   OO  Sales  Qty   OH   OO
------------------------------------------------------------------------------------------------------------------------------------


XXXXXXXXXXXXXXX Xxxxxxxxxxxxxxxxxxxxxxt ZZZ.99 ZZZ.99 XXX XXX ZZ9 ZZZZZ9-ZZZ9-ZZZ9-ZZZ9-ZZZZZ9-ZZZ9-ZZZ9-ZZZ9-ZZZZZ9-ZZZ9-ZZZ9-ZZZ9-
XXXXXXXXXXXXXXX Xxxxxxxxxxxxxxxxxxxxxxt ZZZ.99 ZZZ.99 XXX XXX ZZ9 ZZZZZ9-ZZZ9-ZZZ9-ZZZ9-ZZZZZ9-ZZZ9-ZZZ9-ZZZ9-ZZZZZ9-ZZZ9-ZZZ9-ZZZ9-
XXXXXXXXXXXXXXX Xxxxxxxxxxxxxxxxxxxxxxt ZZZ.99 ZZZ.99 XXX XXX ZZ9 ZZZZZ9-ZZZ9-ZZZ9-ZZZ9-ZZZZZ9-ZZZ9-ZZZ9-ZZZ9-ZZZZZ9-ZZZ9-ZZZ9-ZZZ9-
XXXXXXXXXXXXXXX Xxxxxxxxxxxxxxxxxxxxxxt ZZZ.99 ZZZ.99 XXX XXX ZZ9 ZZZZZ9-ZZZ9-ZZZ9-ZZZ9-ZZZZZ9-ZZZ9-ZZZ9-ZZZ9-ZZZZZ9-ZZZ9-ZZZ9-ZZZ9-
XXXXXXXXXXXXXXX Xxxxxxxxxxxxxxxxxxxxxxt ZZZ.99 ZZZ.99 XXX XXX ZZ9 ZZZZZ9-ZZZ9-ZZZ9-ZZZ9-ZZZZZ9-ZZZ9-ZZZ9-ZZZ9-ZZZZZ9-ZZZ9-ZZZ9-ZZZ9-
XXXXXXXXXXXXXXX Xxxxxxxxxxxxxxxxxxxxxxt ZZZ.99 ZZZ.99 XXX XXX ZZ9 ZZZZZ9-ZZZ9-ZZZ9-ZZZ9-ZZZZZ9-ZZZ9-ZZZ9-ZZZ9-ZZZZZ9-ZZZ9-ZZZ9-ZZZ9-
XXXXXXXXXXXXXXX Xxxxxxxxxxxxxxxxxxxxxxt ZZZ.99 ZZZ.99 XXX XXX ZZ9 ZZZZZ9-ZZZ9-ZZZ9-ZZZ9-ZZZZZ9-ZZZ9-ZZZ9-ZZZ9-ZZZZZ9-ZZZ9-ZZZ9-ZZZ9-
XXXXXXXXXXXXXXX Xxxxxxxxxxxxxxxxxxxxxxt ZZZ.99 ZZZ.99 XXX XXX ZZ9 ZZZZZ9-ZZZ9-ZZZ9-ZZZ9-ZZZZZ9-ZZZ9-ZZZ9-ZZZ9-ZZZZZ9-ZZZ9-ZZZ9-ZZZ9-
XXXXXXXXXXXXXXX Xxxxxxxxxxxxxxxxxxxxxxt ZZZ.99 ZZZ.99 XXX XXX ZZ9 ZZZZZ9-ZZZ9-ZZZ9-ZZZ9-ZZZZZ9-ZZZ9-ZZZ9-ZZZ9-ZZZZZ9-ZZZ9-ZZZ9-ZZZ9-

                                                   Report totals: ZZZZZ9-ZZZ9-ZZZ9-ZZZ9-ZZZZZ9-ZZZ9-ZZZ9-ZZZ9-ZZZZZ9-ZZZ9-ZZZ9-ZZZ9-



                                                        *** End of report ***
```

```
Date 01/25/09  Time 16:40:50                       Camptown Hardware, Inc. DEMO1                   Report# 1801  User MAS  Page 0001

                                                I T E M   S A L E S   B Y   D A T E

Item type: "All"                                    B Y   I T E M   N U M B E R
Item range: "First"         to "Last"                                                 Sales date range:   "Earliest"  to "Latest"
Location: 1     Pine Street                                                           Sales range:        "Lowest"    to "Highest"
Vendor: "All"                                                                         Gross profit range: "Lowest"    to "Highest"
Category/sub-category:   "All"   / "All"                                              Qty sold range:     "Lowest"    to "Highest"
BxGF/seasn/Type/Calib: "All"/"All"/"All"/"All"                                        Qty on order range: "Lowest"    to "Highest"
Corporate file created:  Store-#: 999                                                 Qty on hand range:  "Lowest"    to "Highest"
Local file created: Filename: Xxxxxxxxxxxx

------------------------------------------------------------------------------------------------------------------------------------
Item-#          Description-1       LstCost Price1 Q-PTD Sls-PTD  GP-PTD OnHnd OnOrd Q-YTD  Sls-YTD   GP-YTD Pak Cat/Sub Cl/Sb Gr/Sb
------------------------------------------------------------------------------------------------------------------------------------

XXXXXXXXXXXXXXX Xxxxxxxxxxxxxxxxxxxt ZZZ.99 ZZZ.99 ZZZZ9-ZZZZZZ9-ZZZZZZ9-ZZZZ9-ZZZZ9-ZZZZ9-ZZZZZZZ9-ZZZZZZZ9-ZZ9 XXX/XXX XX/XX XX/XX
XXXXXXXXXXXXXXX Xxxxxxxxxxxxxxxxxxxt ZZZ.99 ZZZ.99 ZZZZ9-ZZZZZZ9-ZZZZZZ9-ZZZZ9-ZZZZ9-ZZZZ9-ZZZZZZZ9-ZZZZZZZ9-ZZ9 XXX/XXX XX/XX XX/XX
XXXXXXXXXXXXXXX Xxxxxxxxxxxxxxxxxxxt ZZZ.99 ZZZ.99 ZZZZ9-ZZZZZZ9-ZZZZZZ9-ZZZZ9-ZZZZ9-ZZZZ9-ZZZZZZZ9-ZZZZZZZ9-ZZ9 XXX/XXX XX/XX XX/XX
XXXXXXXXXXXXXXX Xxxxxxxxxxxxxxxxxxxt ZZZ.99 ZZZ.99 ZZZZ9-ZZZZZZ9-ZZZZZZ9-ZZZZ9-ZZZZ9-ZZZZ9-ZZZZZZZ9-ZZZZZZZ9-ZZ9 XXX/XXX XX/XX XX/XX
XXXXXXXXXXXXXXX Xxxxxxxxxxxxxxxxxxxt ZZZ.99 ZZZ.99 ZZZZ9-ZZZZZZ9-ZZZZZZ9-ZZZZ9-ZZZZ9-ZZZZ9-ZZZZZZZ9-ZZZZZZZ9-ZZ9 XXX/XXX XX/XX XX/XX
XXXXXXXXXXXXXXX Xxxxxxxxxxxxxxxxxxxt ZZZ.99 ZZZ.99 ZZZZ9-ZZZZZZ9-ZZZZZZ9-ZZZZ9-ZZZZ9-ZZZZ9-ZZZZZZZ9-ZZZZZZZ9-ZZ9 XXX/XXX XX/XX XX/XX
XXXXXXXXXXXXXXX Xxxxxxxxxxxxxxxxxxxt ZZZ.99 ZZZ.99 ZZZZ9-ZZZZZZ9-ZZZZZZ9-ZZZZ9-ZZZZ9-ZZZZ9-ZZZZZZZ9-ZZZZZZZ9-ZZ9 XXX/XXX XX/XX XX/XX
XXXXXXXXXXXXXXX Xxxxxxxxxxxxxxxxxxxt ZZZ.99 ZZZ.99 ZZZZ9-ZZZZZZ9-ZZZZZZ9-ZZZZ9-ZZZZ9-ZZZZ9-ZZZZZZZ9-ZZZZZZZ9-ZZ9 XXX/XXX XX/XX XX/XX
XXXXXXXXXXXXXXX Xxxxxxxxxxxxxxxxxxxt ZZZ.99 ZZZ.99 ZZZZ9-ZZZZZZ9-ZZZZZZ9-ZZZZ9-ZZZZ9-ZZZZ9-ZZZZZZZ9-ZZZZZZZ9-ZZ9 XXX/XXX XX/XX XX/XX
XXXXXXXXXXXXXXX Xxxxxxxxxxxxxxxxxxxt ZZZ.99 ZZZ.99 ZZZZ9-ZZZZZZ9-ZZZZZZ9-ZZZZ9-ZZZZ9-ZZZZ9-ZZZZZZZ9-ZZZZZZZ9-ZZ9 XXX/XXX XX/XX XX/XX

Report totals:                                     ZZZZ9-ZZZZZZ9-ZZZZZZ9-ZZZZ9-ZZZZ9-ZZZZ9-ZZZZZZZ9-ZZZZZZZ9-




                                                        -- End of report --
```

```
Date 01/25/09  Time 16:44:17                       Camptown Hardware, Inc. DEMO1                   Report# 1802  User MAS  Page 0001

                                           I T E M   S A L E S   Y E A R - T O - D A T E

Item type: "All"                                    B Y   I T E M   N U M B E R
Item range: "First"         to "Last"
Location: 1     Pine Street                                                           Sales range:        "Lowest"    to "Highest"
Vendor: "All"                                                                         Grs profit % range: "Lowest"    to "Highest"
Category/sub-category:   "All"   / "All"                                              Qty sold range:     "Lowest"    to "Highest"
BxGF/seasn/Type/Calib: "All"/"All"/"All"/"All"                                        Qty on order range: "Lowest"    to "Highest"
Corporate file created:  Store-#: 999                                                 Qty on hand range:  "Lowest"    to "Highest"
Local file created: Filename: Xxxxxxxxxxxx

------------------------------------------------------------------------------------------------------------------------------------
Item-#          Description-1         LstCost Price1 Q-YTD  Sls-YTD GP%YTD OnHnd OnOrd Q-LYR  Sls-LYR GP%LYR Pak Cat/Sub Cl/Sb Gr/Sb
------------------------------------------------------------------------------------------------------------------------------------

XXXXXXXXXXXXXXX Xxxxxxxxxxxxxxxxxxxxxt ZZZ.99 ZZZ.99 ZZZZ9-ZZZZZZZ9-ZZZZZ9-ZZZZ9-ZZZZ9-ZZZZ9-ZZZZZZZ9-ZZZZZ9-ZZ9 XXX/XXX XX/XX XX/XX
XXXXXXXXXXXXXXX Xxxxxxxxxxxxxxxxxxxxxt ZZZ.99 ZZZ.99 ZZZZ9-ZZZZZZZ9-ZZZZZ9-ZZZZ9-ZZZZ9-ZZZZ9-ZZZZZZZ9-ZZZZZ9-ZZ9 XXX/XXX XX/XX XX/XX
XXXXXXXXXXXXXXX Xxxxxxxxxxxxxxxxxxxxxt ZZZ.99 ZZZ.99 ZZZZ9-ZZZZZZZ9-ZZZZZ9-ZZZZ9-ZZZZ9-ZZZZ9-ZZZZZZZ9-ZZZZZ9-ZZ9 XXX/XXX XX/XX XX/XX
XXXXXXXXXXXXXXX Xxxxxxxxxxxxxxxxxxxxxt ZZZ.99 ZZZ.99 ZZZZ9-ZZZZZZZ9-ZZZZZ9-ZZZZ9-ZZZZ9-ZZZZ9-ZZZZZZZ9-ZZZZZ9-ZZ9 XXX/XXX XX/XX XX/XX
XXXXXXXXXXXXXXX Xxxxxxxxxxxxxxxxxxxxxt ZZZ.99 ZZZ.99 ZZZZ9-ZZZZZZZ9-ZZZZZ9-ZZZZ9-ZZZZ9-ZZZZ9-ZZZZZZZ9-ZZZZZ9-ZZ9 XXX/XXX XX/XX XX/XX
XXXXXXXXXXXXXXX Xxxxxxxxxxxxxxxxxxxxxt ZZZ.99 ZZZ.99 ZZZZ9-ZZZZZZZ9-ZZZZZ9-ZZZZ9-ZZZZ9-ZZZZ9-ZZZZZZZ9-ZZZZZ9-ZZ9 XXX/XXX XX/XX XX/XX
XXXXXXXXXXXXXXX Xxxxxxxxxxxxxxxxxxxxxt ZZZ.99 ZZZ.99 ZZZZ9-ZZZZZZZ9-ZZZZZ9-ZZZZ9-ZZZZ9-ZZZZ9-ZZZZZZZ9-ZZZZZ9-ZZ9 XXX/XXX XX/XX XX/XX
XXXXXXXXXXXXXXX Xxxxxxxxxxxxxxxxxxxxxt ZZZ.99 ZZZ.99 ZZZZ9-ZZZZZZZ9-ZZZZZ9-ZZZZ9-ZZZZ9-ZZZZ9-ZZZZZZZ9-ZZZZZ9-ZZ9 XXX/XXX XX/XX XX/XX
XXXXXXXXXXXXXXX Xxxxxxxxxxxxxxxxxxxxxt ZZZ.99 ZZZ.99 ZZZZ9-ZZZZZZZ9-ZZZZZ9-ZZZZ9-ZZZZ9-ZZZZ9-ZZZZZZZ9-ZZZZZ9-ZZ9 XXX/XXX XX/XX XX/XX
XXXXXXXXXXXXXXX Xxxxxxxxxxxxxxxxxxxxxt ZZZ.99 ZZZ.99 ZZZZ9-ZZZZZZZ9-ZZZZZ9-ZZZZ9-ZZZZ9-ZZZZ9-ZZZZZZZ9-ZZZZZ9-ZZ9 XXX/XXX XX/XX XX/XX

Report totals:                                       ZZZZ9-ZZZZZZZ9-       ZZZZ9-ZZZZ9-ZZZZ9-ZZZZZZZ9-

                                                        -- End of report --
```

```
Date 99/99/99  Time 99:99:99                          Camptown Hardware, Inc.                      Report# 9999  User XXX  Page 9999

                                M U L T I - L O C A T I O N   S T O C K   S T A T U S   R E P O R T

 Item type: Xxxxxxxxxxxxxxxx              XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX    Sales range: ZZ,ZZZ,ZZ9- to ZZ,ZZZ,ZZ9-
Item range: XXXXXXXXXXXXXXX to XXXXXXXXXXXXXXX                                      Qty sold date range: 99/99/99xx to 99/99/99
 Locations: XXXXX XXXXX XXXXX XXXXX                                                      Qty sold range: ZZ,ZZZ,ZZ9- to ZZ,ZZZ,ZZ9-
    Vendor: XXXXXXXXXXXX Xxxxxxxxxxxxxxxxxxxxxxxxx                                   Qty on order range: ZZ,ZZZ,ZZ9- to ZZ,ZZZ,ZZ9-
  Category/sub-category: XXXXXxx / XXXXXxx                                           Qty on hand range: ZZ,ZZZ,ZZ9- to ZZ,ZZZ,ZZ9-
user1/user2/user3/user4: XXXXX/XXXXX/XXXXX/XXXXX                                Corporate file created:  Store-#: 999
                                                                                    Local file created: Filename: Xxxxxxxxxxxx

------------------------------------------------------------------------------------------------------------------------------------
Item-#          Description-1               |-----Loc:XXXX------| |-----Loc:XXXX------| |-----Loc:XXXX------| |-----Grp:XXXXX-----|
                    Cost  Price Sub Cls Pak    Sls  Qty   OH   OO    Sls  Qty   OH   OO    Sls  Qty   OH   OO    Sls  Qty   OH   OO
------------------------------------------------------------------------------------------------------------------------------------

XXXXXXXXXXXXXXX Xxxxxxxxxxxxxxxxxxxxxxxxx
                  ZZZ.99 ZZZ.99 XXX XXX ZZ9 ZZZZZ9-ZZZ9-ZZZ9-ZZZ9-ZZZZZ9-ZZZ9-ZZZ9-ZZZ9-ZZZZZ9-ZZZ9-ZZZ9-ZZZ9-ZZZZZ9-ZZZ9-ZZZ9-ZZZ9-
XXXXXXXXXXXXXXX Xxxxxxxxxxxxxxxxxxxxxxxxx
                  ZZZ.99 ZZZ.99 XXX XXX ZZ9 ZZZZZ9-ZZZ9-ZZZ9-ZZZ9-ZZZZZ9-ZZZ9-ZZZ9-ZZZ9-ZZZZZ9-ZZZ9-ZZZ9-ZZZ9-ZZZZZ9-ZZZ9-ZZZ9-ZZZ9-
XXXXXXXXXXXXXXX Xxxxxxxxxxxxxxxxxxxxxxxxx
                  ZZZ.99 ZZZ.99 XXX XXX ZZ9 ZZZZZ9-ZZZ9-ZZZ9-ZZZ9-ZZZZZ9-ZZZ9-ZZZ9-ZZZ9-ZZZZZ9-ZZZ9-ZZZ9-ZZZ9-ZZZZZ9-ZZZ9-ZZZ9-ZZZ9-
XXXXXXXXXXXXXXX Xxxxxxxxxxxxxxxxxxxxxxxxx
                  ZZZ.99 ZZZ.99 XXX XXX ZZ9 ZZZZZ9-ZZZ9-ZZZ9-ZZZ9-ZZZZZ9-ZZZ9-ZZZ9-ZZZ9-ZZZZZ9-ZZZ9-ZZZ9-ZZZ9-ZZZZZ9-ZZZ9-ZZZ9-ZZZ9-
XXXXXXXXXXXXXXX Xxxxxxxxxxxxxxxxxxxxxxxxx
                  ZZZ.99 ZZZ.99 XXX XXX ZZ9 ZZZZZ9-ZZZ9-ZZZ9-ZZZ9-ZZZZZ9-ZZZ9-ZZZ9-ZZZ9-ZZZZZ9-ZZZ9-ZZZ9-ZZZ9-ZZZZZ9-ZZZ9-ZZZ9-ZZZ9-
XXXXXXXXXXXXXXX Xxxxxxxxxxxxxxxxxxxxxxxxx
                  ZZZ.99 ZZZ.99 XXX XXX ZZ9 ZZZZZ9-ZZZ9-ZZZ9-ZZZ9-ZZZZZ9-ZZZ9-ZZZ9-ZZZ9-ZZZZZ9-ZZZ9-ZZZ9-ZZZ9-ZZZZZ9-ZZZ9-ZZZ9-ZZZ9-

                             Report totals: ZZZZZ9-ZZZ9-ZZZ9-ZZZ9-ZZZZZ9-ZZZ9-ZZZ9-ZZZ9-ZZZZZ9-ZZZ9-ZZZ9-ZZZ9-ZZZZZ9-ZZZ9-ZZZ9-ZZZ9-


                                                        *** End of report ***
```