



## **Enterprise Purge**

**Infinity Software Solutions, Inc**  
1414 19<sup>th</sup> Street  
Palm Harbor, FL 34683  
727-785-8563  
infinity\_software@verizon.net

Date: 5/2/2014

### **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**

Files to the customer: sy\SYPRGU.GNT      Extract program

Confidential

Date: 5/2/2014

### 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  XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
2. Company-id    XXXXXXXXXX  XXXXXXXXXXXXXXXXXXXXXXXX
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

Files to the customer:	ar\ARCUSL.GNT	Customer lookup program
	sa\SAMOH1.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

Date: 3/26/2014

### 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  XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
2. Company-id    XXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXX
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:

<u>Filename</u>	<u>Description</u>	<u>Notes</u>
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.

**Files affected by the enhancement**

Files to the customer: sy\SYPRGO.GNT	Extract program parameter entry program
sy\SYPRGU.GNT	Extract program
SYMENU.DAT	Menu definition file

Confidential