

## Introduction to PerSay for OpenQM

# PerSay Multivalue Software Design system for OpenQM

## Introduction

### ***What is PerSay?***

Designed by and for software developers working to build high function multivalue database applications, PerSay is a system of layout and BASIC code generation for creating entry screens and reports. Originally created in 1991, PerSay has evolved into a feature rich system, proven over years of use. The 1.2 version released in 2004 is specifically tuned and optimized for Ladybridge's OpenQM database product. OpenQM is a robust and complete multiuser, multivalue database system for Linux and Windows.

As quoted from <http://www.openqm.org>

“OpenQM is a fork of the 'QM' database. QM is a commercial MultiValue database modeled after Prime Information. QM was developed by Ladybridge Systems in Northampton, England. QM is not based on, and has no code included from other MultiValue databases.”

### ***Why do we need PerSay?***

After an installation of OpenQM, a new user is taken to a command prompt and asking themselves “OK, now what do I do?”. The fulfillment of an industrial strength multiuser business database system is not at all evident to such a user. PerSay offers immediate gratification in showing how entry screens, reports and the integrated DICT editor “Dr.Dict” are used to easily build business software right now.

Conversely, any of the very experienced multivalue software developers will find PerSay a joy to use. Complete business systems can be built in hours instead of weeks. Having PerSay provide the robust framework for software allows a developer to focus on the creative work of design, rather than having to hand code every line. The consistency and easily discerned flow of the instantly generated BASIC code provide the developer an advantage.

### ***What's in it?***

There are three main tools in PerSay.

- 1) An visual entry (or inquiry) screen layout and BASIC code generator. The developer can choose either to generate entry screens in a fully cursor/edit key aware modern user interface or the ancient INPUT style. A PerSay generated BASIC program is generated to be instantly compiled into a business class program.
- 2) A visual report layout with a Point-and-Shoot editor that understands how OpenQM's query language works, building a front end for user's to enter report parameters (like a date range, Customer ID, etc.) and then generate BASIC code for both the front end screen and the OpenQM query itself sent to Screen or Printer.
- 3) An integrated Dictionary editor “Dr. Dict” that is tied to both the screen and report layout tools and a valuable standalone tool for managing the important DICT items of an OpenQM database.

## **Introduction to PerSay for OpenQM**

### ***What is it not?***

Graphic User Interface. PerSay is strictly a text screen format while fully cursor / edit key based, it does not require a user to leave the keyboard to find a mouse, find the keyboard, find a mouse, etc.

### ***What does PerSay require?***

Only a working OpenQM system. No run-time modules or anything other than the BASIC code PerSay generates is needed. For the cursor/edit key aware modern user interface to function, PerSay supplied file for Terminal Definitions (PER.TERM) must be present.

### ***How is PerSay sold or licensed?***

PerSay is provided free of charge using the same Open Source licensing that OpenQM does, the GNU Public License (GPL). In summary this means you can use it free, but may not call it your own or sell it as your own product. No warranty, fitness for a purpose or support is expressed or implied.

### ***Contacting Performant Systems***

Based in the Los Angeles area, Performant Systems is a computer software design company, providing design services in Multivalue database, PHP/MySQL web based applications, Linux systems, training and technical authoring. Makers of the ExpoGrid Exhibition Management system. Providers of AccuTerm GUI Development Classes. Developers of PerSay Multivalue database design software. PerSay and ExpoGrid are Trademarks of Performant Systems.

PerSay is developed by Alan Gruskoff

Email: [linux@performantsystems.com](mailto:linux@performantsystems.com)

Phone: (310) 356-6839

Web Site: <http://performantsystems.com>

# Introduction to PerSay for OpenQM

## *Installation for Linux*

The distribution item persay\_1\_2.tar.gz contains two directories: persay and persay.demo. Installation is mostly a matter of unpacking these two directories and copying (or moving) them to within the same directory that you installed OpenQM in, typically /usr/qmsys.

You should wind up with these two directories:

```
/usr/qmsys/persay
/usr/qmsys/persay.demo
```

Once these directories are in place, login to the OpenQM system in the qmsys directory, ex:

```
cd /usr/qmsys
qm
```

Then, let OpenQM know about your two new Accounts by editing the file that is the register for Accounts amazingly called ACCOUNTS. Using the SED or ED editors as you like, make two new items in ACCOUNTS file such as:

```
:SED ACCOUNTS PERSAY PERSAY.DEMO
```

Pay careful attention to Case and spelling, exactly as indicated. Make two new items in the ACCOUNTS file that look like this:

```
PERSAY
line 1:  @QMSYS/persay

PERSAY.DEMO
line 1:  @QMSYS/persay.demo
```

The token @QMSYS is a pointer to wherever the home folder of QM is.

**In the SED editor use <Ctrl>XW to Write an item, use <Ctrl>XC to exit.**

You can now LOGTO PERSAY.DEMO at an OpenQM prompt or from a Linux command line do:

```
cd /usr/qmsys/persay.demo
qm
```

# Introduction to PerSay for OpenQM

## *Installation for Windows*

The distribution item PerSay\_OpenQM\_Win.zip contains two directories: persay and persay.demo. Installation is mostly a matter of unzipping these two directories and copying (or moving) them to within the same directory that you installed OpenQM in, typically C:\QMSYS.

You should wind up with these two directories:

```
C:\QMSYS\persay
C:\QMSYS\persay.demo
```

Once these directories are in place, login to the OpenQM system in the QMSYS directory.

Then, let OpenQM know about your two new Accounts by editing the file that is the register for Accounts amazingly called ACCOUNTS. Using the SED or ED editors as you like, make two new items in ACCOUNTS file such as:

```
:SED ACCOUNTS PERSAY PERSAY.DEMO
```

Pay careful attention to Case and spelling, exactly as indicated. Make two new items in the ACCOUNTS file that look like this:

```
PERSAY
line 1:  @QMSYS\persay

PERSAY.DEMO
line 1:  @QMSYS\persay.demo
```

The token @QMSYS is a pointer to wherever the home folder of QM is.

**In the SED editor use <Ctrl>XW to Write an item, use <Ctrl>XC to exit.**

You can now LOGTO PERSAY.DEMO at an OpenQM prompt.

# Introduction to PerSay for OpenQM

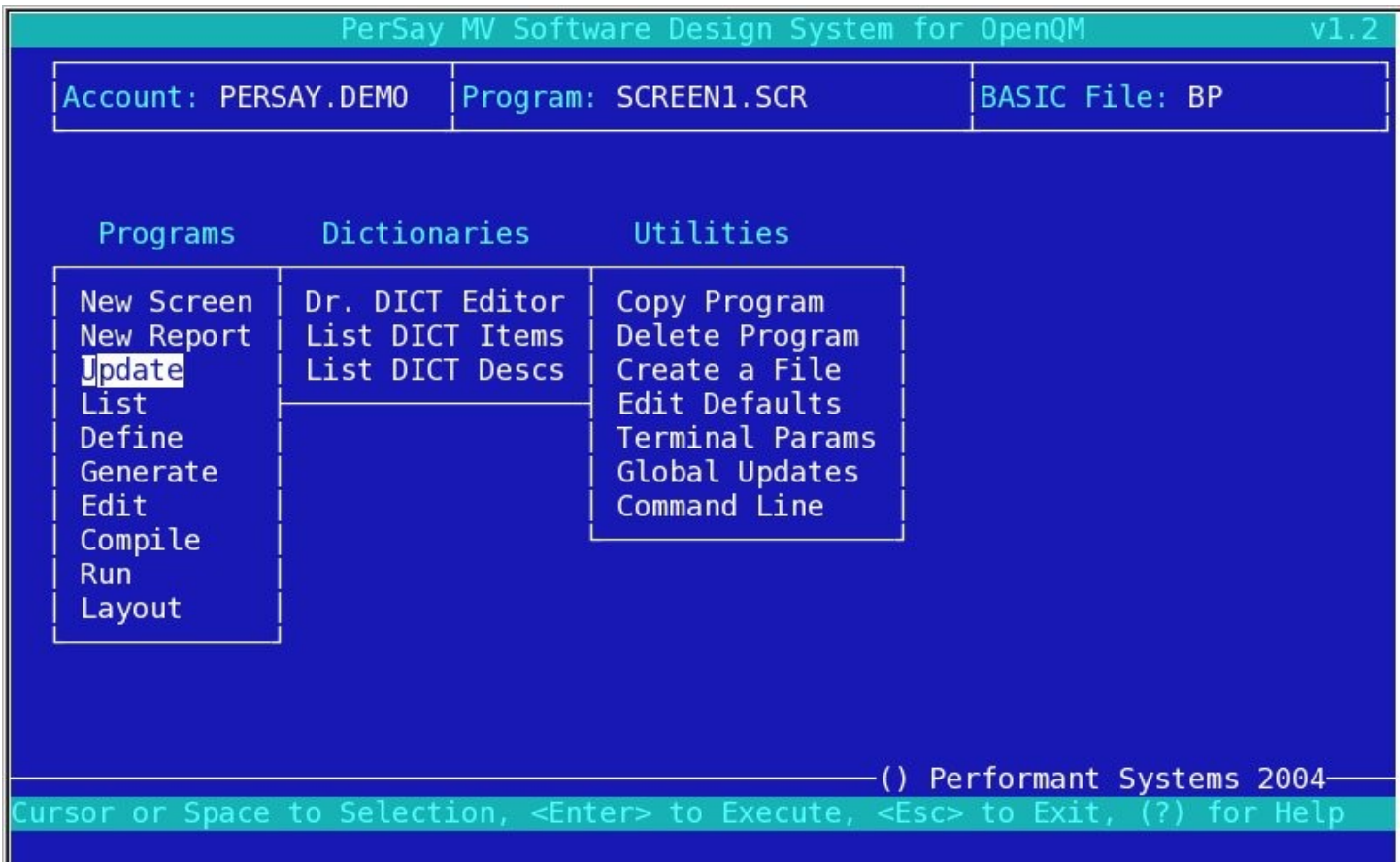
## Which Account is which?

The PERSAY account is used to hold the programs for the PerSay systems and the PER.TERM file for Terminal Definitions. It is consider a static software that is easily setup for use in a working production account. We don't do development in the PERSAY account, we do development in a live or development account. Only LOGTO PERSAY to let it setup another account to have PerSay available.

The PERSAY.DEMO account is as you might imagine a demonstration account that is meant to allows learning and experimentation. Feel free to do what ever you like in the PERSAY.DEMO account, it can always be restored from the distribution item.

## The PerSay Menu

PerSay is run at a command line as: RUN PER.BP PER.MENU or just PER.MENU.



## Built-in Help - (?) for help

With the highlighted pointer on any of the PerSay Menu Selections, hit (?) for a description of what that Selection does.

## Introduction to PerSay for OpenQM

### ***Using PerSay in a live production environment***

Let's say you have a working account on your OpenQM system called BUSINESS that you would like to be able to use PerSay within. Easily done. LOGTO PERSAY, you are first presented with an installation dialog, rather than the PerSay Menu. Just tell PerSay the name of the account to install into (BUSINESS) and it does it all for you. Installation is essentially writing all the needed Q-pointers to PERSAY into the BUSINESS account.

Note that you do not need the PERSAY or PERSAY.DEMO accounts present to run any programs that PerSay has generated for you as they are just pure BASIC programs. You DO need to have a PER.TERM file available to your account if you are running a cursor based (CHAR) style entry screen or report. Also the PER.INC file must be available to do any BASIC compile of a PerSay generated program. Both the PER.TERM and PER.INC files can be copied to a local account, if needed. Keeping the PERSAY account takes care of all of that for you.

### ***Starting out in PerSay***

You can LOGTO PERSAY.DEMO at an OpenQM prompt or from a Linux command line do:

```
cd /usr/qmsys/persay.demo
qm
```

Cursor or <Spacebar> to move the Menu item highlight to "List", hit <Enter>. We see the below Selected List Window, as is used throughout PerSay, here offering selections to items in the PER.SCREENS file.

Note that all PerSay Layout models are kept in items in the PER.SCREENS file. In the PERSAY.DEMO account the PER.SCREENS is local to that account. The System Admin for OpenQM may choose to keep this file in whatever account is convenient, likely the same account that the programs file (usually called BP) is kept in. PerSay will create any file for you with the "Create a File" menu option.

# Introduction to PerSay for OpenQM

```
PerSay MV Software Design System for OpenQM v1.2
Account: PERSAY.DEMO Program: REPLIST.RPT BASIC File: BP
PER.SCREENS Selected List Window
Line# @ID.....
A> CUSTOMERS.RPT
B> REP1.SCR
C> REP2.SCR
D> REPLIST.RPT
E> SCREEN1.SCR
F> ZIPS.SCR
N
N
U
L
D
G
E
C
R
L
Type Line# Letter (A-F) or <Esc>: [ ]
() Performant Systems 2004
```

In the PER.SCREENS file in the PERSAY.DEMO account, all the entry screen models are items that end in “.SCR”, all the report models are items that end in “.RPT”. These extension are NOT required, just shown for ease of viewing. You can name PerSay models anything reasonable, PerSay knows which are screens or reports. Use whatever naming convention you like. Hit “E” to choose the SCREEN1.SCR item.



## Introduction to PerSay for OpenQM

As the Entry Screen Model Editor indicates, you can hit <Esc>1 for help, meaning hit the <Esc> key, then the 1 key. Even if your Term Type is mis-set, this key sequence will always work. Here is the result of <Esc>1

```
PerSay MV Software Design System for OpenQM v1.2
Function key definitions:
F1 | F2 | F3 | F4 | F5 | F6 | F7 | F8 | F9 | F10
Esc 1 | Esc 2 | Esc 3 | Esc 4 | Esc 5 | Esc 6 | Esc 7 | Esc 8 | Esc 9 | Esc 0
Help | Delete | Delete | Insert | Add | Update | Delete | Next | Move | Save
      | Char  | Line   | Line   | Attr  | Attr  | Attr  | Attr | Element | & Exit
                                           | Esc L | Esc R | Esc S | Esc A
                                           | Clear | ReOrder | screen | Attr
                                           | Line  | Attr   | Redraw | List
Use the cursor keys to point to where to edit.
Hit <Tab> to move 10 spaces right.
Hit <Shift><Tab> or <Ctrl>B to move 10 spaces left.
<Esc> at any dialog line prompt to exit.
<Enter> to continue:
```

As you can see, Function keys 1 thru 10 are synonymous with hitting <Esc> 1 thru 0, either way is fine. Take note of these important navigation features, certainly exiting with <Esc>0 or F10.

One feature that makes it fast and easy to get around is the F8 – Next Attr key. Starting always with the first screen field, each successive hit of F8 moves the cursor to the next field.

## Introduction to PerSay for OpenQM

With the cursor on ANY part of an underscore field marker, hit F6 – Update Attr to make updates to an existing field as shown here for the first field: ID

```
PerSay Demo - Customer Entry Screen [SCREEN1.SCR]
Customer ID: _____ <Esc> to exit
                           <Ctrl>N toggles Insert
Attribute Parameters Updates
  DICT ID: ID                AMC: 0
1) Conversion Format...
2) Justification (L/R): L
3) Width on screen....: 7
4) Attribute Type Code: E
5) (M)ultiple (S)ingle: S
6) Multivalue Set.....:
7) Capitals (Y/N).....: N
8) Required (Y/N).....: N
9) Code Validations...:
10) Pattern Matching...:
11) Default Value.....:
12) Find List Attribs...: ID, COMPANY, CONTACT
13) Code Before Entry...:
14) Code After Entry...:
15) Display Calculation:
16) Translate ID...:
17) Translate File:
18) Translate AMC.:
19) MV Depth...:
20) MV Spacing:
21) MV Properties:
Change (1-21) or <Enter> to continue: [ ]__
```

This Attribute Parameters Updates screen is called by the F6 key when the cursor is on any underscore of a field.

As you notice there are 21 entries for the way this field should operate. Here is a simple ID entry that adds the deluxe feature of integrating the PER.FIND.LIST Selected List Window simply by entering a comma separated list of DICT items for the primary file, always with the ID or @ID item first. The first item in the list is considered the ID, it can be named anything you like.

► Don't make an initial ID Attribute Required or the user can never exit! A null entry will exit the program.

► Do make most codes or Y/N entries as Capitals (Y) as OpenQM is case sensitive. Meaning “y” is NOT equal to “Y”.

The super easy way to validate a Y/N entry is with 9) Code Validations, just enter a comma separated list without quotes like: Y,N A,B,C etc.

## Introduction to PerSay for OpenQM

Back at the Entry Screen Model Editor, hit F8 until the cursor is on an underscore of the Zip Code field, then hit F6 to update that field. This field has significant custom code for a lookup that is a good example of how this is done.

```
PerSay Demo - Customer Entry Screen [SCREEN1.SCR]
Customer ID: _____ <Esc> to exit
                        <Ctrl>N toggles Insert
Attribute Parameters Updates
DICT ID: ZIP          AMC: 6
1) Con | Ln# Source Code.....
2) Jus | 1> IF INDATA = "?" THEN
3) Wid | 2> SELECTION = \ BY CITY\
4) Att | 3> CALL PER.FIND.LIST(LIST.ID, "ZIPS", SELECTION, "", "@ID, CITY,
5) (M) | 4> GOSUB SHOW.SCREEN.MASK
6) Mul | 5> ** Note: R E T R Y is a PerSay token word to do the entry
7) Cap | 6> IF LIST.ID = "" THEN RETRY
8) Req | 7> ENTRY = LIST.ID
9) Cod | 8> ITEM(AMC) = LIST.ID
10) Pat | 9> GOSUB SHOW.SCREEN.DATA
11) Def | 10> END
12) Fin | 11> **
13) Cod | 12> ZIPS.ITEM = ""
14) Cod |
15) Display Calculation:
Change (1-24), (A)dd, (D)elete, (I)nsert, (C)opy, Page(+n/-n) or <Enter>: [ ]
```

This custom code points out some important features about how the entry process works.

To start with a variable called ENTRY is set to an initial value, usually the contents of the field we are working on. Default values are set automatically here or you can code your own in the Code Before Entry section.

Next the GET.ENTRY routine handles what keys the user hit, returning with a variable called INDATA for what the user keyed in and return the variable ENTRY with the results which are null on a non-valid entry or contain the field contents on a valid entry. In the above example, we see that PerSay traps out an initial (?) and returns that as INDATA. PerSay also traps out any navigational keys like <Esc>, <Tab> and it's opposite <Shift><Tab>, all of which are handled automatically within the GET.ENTRY routine.

PerSay adds one and only one thing to the normal OpenQM BASIC code: the RETRY token. When the code generator sees a RETRY in the custom code, it replaces that token with "GOTO the same entry" code. Useful on validations.

Also useful in validations is the ever present SHOW.ERR.MSG routine built the GET.ENTRY routine contained always within the \$INCLUDE PER.INC PER.CHAR.ENTRY item. Set a variable ERR.MSG

## Introduction to PerSay for OpenQM

to a up to 68 chars of a text message, then GOSUB SHOW.ERR.MSG.  
Again, look at the examples and try some for yourself.

The above Attribute Parameters Updates screen is for how each field works. PerSay also offers a screen to update features or properties for the program itself. From the PerSay menu, select "Define".

```
PerSay MV Software Design System for OpenQM v1.2
Account: PERSAY.DEMO Program: SCREEN1.SCR BASIC File: BP
PerSay Program Definition v1.2
1) Primary Filename...: CUSTOMERS
2) Inquiry Only?.....: N
3) Require On File?...: N
4) Allow Deletes?....: Y
5) Create (NEW) IDs?..: Y      6) IDs File: DICT CUSTOMERS  7) AMC: 2
8) Is Subroutine?....: N
9) Passing Parameters:
10) Next screen Code...:
11) Setup Time Code...: OPEN "ZIPS" TO ZIPS ELSE ERR.MSG = "NO ZIPS FILE."
12) After Read Code...: READ ZIPS.ITEM FROM ZIPS,ITEM(6) ELSE ZIPS.ITEM =
13) Pre File Time Code:
14) Post File Time....:
15) Delete Time Code...:
16) Exit Time Code....:
17) Print Time Code...:      18) Fax Code:
Change (1-16), (F)ile or <Esc> to exit: __
```

Here the developer can make entries as to how the generated program works.

Note the 5) Create (NEW) IDs entry. If set to (Y)es, PerSay wants to know where/what data item is to be auto-incremented when the User enters "NEW" for an ID. This can be in a file or a DICT of a file as seen in field 6). The attribute location or (AMC for Attribute Mark Count) is preferred to be 2 as PerSay will write an "X" in AMC 1 to indicate this is an e(X)tra type attribute.

Consider looking at entries 11) Setup Time Code and 12) After Read Code to see examples of how your custom code is integrated and kept with the PerSay screen model item.

## Introduction to PerSay for OpenQM

### Where does all this code go?

As most of us learn best by visual examples, just try it. Place some kind of BASIC code in any of the Program Definitions field including comments, then “Generate” that code, and “Edit” that code to see where your custom code entries landed. Importantly, note the Labels that PerSay always generates for an entry screen and how you can rely on these labels to take advantage of them. The PRE.READ: and POST.READ: labels can be very useful places land as are labels SHOW.SCREEN.MASK:, SHOW.SCREEN.DATA: and FILE.TIME:.

Within the Entry Screen Model Editor, the <Esc>A - Attr List function will show a screen like this:

```
PerSay MV Software Design System for OpenQM v1.2
PerSay Attribute Listing for: SCREEN1.SCR Data Filename: CUSTOMERS
POS  ATTR NAME          AMC      JS  WID TYPE... MV SET POSITION CAPS REQ
 1   ID                  0        L   7  ENTRY  S   23,02  N   N
 2   COMPANY            1        L  25  ENTRY  S   23,04  N   N
 3   CONTACT            8        L  20  ENTRY  S   23,06  N   N
 4   ADDR1              2        L  30  ENTRY  S   23,08  N   N
 5   ZIP                6        L  10  ENTRY  S   23,10  N   N
 6   CITY              4        L  20  ENTRY  S   45,10  N   N
 7   STATE             5        L   3  ENTRY  S   74,10  Y   N
 8   PHONE            41        L  13  ENTRY  S   23,12  N   N
 9   FAX              44        L  13  ENTRY  S   23,14  N   N
10   REP.ID            26        L   6  ENTRY  S   23,16  N   N
11   REP.NAME         99        L  20  DISPLAY S   41,16  N   N
12   NOTES            14        L  50  ENTRY  M* 1  23,18  N   N
<Enter> to continue: 
```

Note the order and properties of the listed fields.

Should you need to change the entry order you have two choices, 1) delete and re-enter a field 2) Use <Esc>R – Reorder Attr. PerSay will ask which Attr (field) to move and then what Attr number to move it AFTER.

## Introduction to PerSay for OpenQM

### Report Model Entry

On to designing reports. From the PerSay Menu, select “List” then hit “A” to work with the CUSTOMERS.RPT model item.

Note the prompt for (D)r. Dictionary. Throughout PerSay, there are direct link to call the full featured DICT editor and then return to where you where. Any field that prompts for a DICT item ID will allows a (?) to list those DICT items, so the developer need not break out of the current function to see what DICT items are available.

```
PerSay - Report Model Entry v1.2
Account: PERSAY.DEMO Report: CUSTOMERS.RPT BASIC File: BP
1) Primary File: CUSTOMERS
2) Report Title: Customer Master List
3) Description.: A list of customer Companies, Contacts and last Invoices
4) ID Supress...: N      5) Detail Supress: N      6) Double Space: N
7) Sort Attr 1.: COMPANY      A/D: A      EXP: N
   Sort Attr 2.:              A/D:      EXP:
   Sort Attr 3.:              A/D:      EXP:
   Sort Attr 4.:              A/D:      EXP:
8) Pre-set Selection: WITH REP.NAME
9) User Parameter Input Prompts [on next screen]
10) Report Model Entry      Column Width : 101
COMPANY      CONTACT      ZIP      STATE      REP.NAME
LAST.INV.AMT      LAST.INV.DATE
Change (1-10), (D)r. Dictionary, (F)ile, or <Esc> to exit: [ ]_
```

Here we notice 10 different entries for parameters that will tell PerSay what kind of report front end and query to build. By “front end” we mean the user interface where we can prompt for various parameters like a date range or customer ID. PerSay generates BASIC code to build a front end program that will then execute an OpenQM query sending output to the user's selection of (S)creen or (P)rinter.

On AccuTerm or other quality emulations the Screen option will change video to a 132 column mode if a report is know to be more than 80 columns of output. This is known NOT to work under the standard Linux KDE Konsole. See the Terminal Params screen for the ASCII codes PerSay will send for 132 and 80 column video modes.

Up to four levels of Sort Attributes are available, each has a property for (A)scending/(D)escending and EXPloded for multivalued sorting. The Pre-set Selection means that this part a a query is ALWAYS

## Introduction to PerSay for OpenQM

included first in building a Select statement. Example: WITH BALANCE NE "0"

### 9) User Parameter Input Prompts

```
PerSay - Report Model Entry v1.2
Account: PERSAY.DEMO Report: CUSTOMERS.RPT BASIC File: BP
1) Primary File: CUSTOMERS
2) Report Title: Customer Master List
3) Description.: A list of customer Companies, Contacts and last Invoices
4) ID Supress...: N 5) Detail Supress: N 6) Double Space: N
User Parameter Input Prompts
Prompt# Prompt Screen Message..... OP Format [ ] Prompt Attribute
1. Enter a Company name to list or (ALL)___ = Y COMPANY
2. Enter a ZIP code to list or (ALL) = N ZIP
3.
4.
5.
6.
7.
8.
Enter the Prompt message to be displayed, optional Keyword "(ALL)"
```

Here, the developer has up to 8 opportunities to define user prompts to collect data that will be used to form a query.

#### **Date Ranges**

Here is a hot tip. If you know you are going to need a date range, enter those prompts first, such as

```
Enter starting Date GE D4- N DATE
Enter ending Date LE D4- N DATE
```

PerSay always generates a commented statement such as:

```
** SENTENCE := SPACE(20):\From: \:ENTRY1:\ Thru: \:ENTRY2:\'L'\
```

Just UN-comment that line after the code is generated, and you have instant header output of the range. Users like to know what they are looking at and why.

# Introduction to PerSay for OpenQM

## Building a Report Model by visual layout

```
PerSay - Report Model Entry v1.2
Account: PERSAY.DEMO Report: CUSTOMERS.RPT BASIC File: BP

1) Primary File: CUSTOMERS

CUSTOMERS DICT items.....
@ID          ADDR1          ADDR2          CITY          COMPANY
CONTACT      CUST.ID        FAX            ID            LAST.INV.AMT
LAST.INV.DATE NOTES          PHONE         REP.ID        REP.NAME
STATE        TITLE          TYPE          ZIP

Report Model          Column Total: 101  Displayed from Column: 27
...3....|...4....|...5....|...6....|...7....|...8....|...9....|...0....|
2.          3.    4.  5.          6.          7.
CONTACT          ZIP ST REP.NAME          LAST.INV.AMT LAST.INV.DATE
XXXXXXXXXXXXXXXXXXXX XXXXX XX XXXXXXXXXXXXXXXXXXXX 9,999,999.99 10-29-2004

Change Col# (1-7), (A)dd, (D)el, (I)ns, (W)id, Page(+n/-n) or <Enter>: [ ]__
```

One of the original reasons to create PerSay was to make a way to see how many columns across a report was. This basic feature still holds value in the manner in which a Report Model is built. Simply use your cursor keys to point to a DICT item that is NOT highlighted, hit <Enter> and it's added to the Report as the new last column. Those DICT items that are already used are displayed as highlighted and the cursor will not go to those items.

The Report Model editor then offers to options to Add, Delete or Insert DICT items as needed. The (W)id prompt will toggle 80/132 column modes, if your emulation can handle it. Navigation horizontally is done with the Page + or - options, +3 moves 3 fields right, -4 moves 4 fields left and so on. Just + or - by themselves move the visual model one field right and left respectively.

### Setting up Break-On and Total-On properties

While not perhaps obvious, the Change Col option handles these properties. Enter the Column number, then (Y) for Break-On or (N) to progress to the Total-On prompt (Y/N). PerSay will show a "B" the visual layout for those fields that are set for Break-On, and show a "T" for those fields that are Total-On. Be sure you have a sort defined for the Break-On field and have set Total-On for a numeric field it makes sense to Total-On.

## Introduction to PerSay for OpenQM

### How PerSay Reports work

Since OpenQM has a rich database query language as an integral part of the product, PerSay provides a graceful shell around that robust query language. After any ant user parameters are entered and output device selected, PerSay builds and executes a “SSELECT” (Sort and Select) statement based on the report model and user inputs. Then a “LIST” statement is executed (if there is something to list) based on the visual model layout.

The developer may choose to add specific statements to embellish these statements if necessary, realizing that when the program is again “Generate”d, that those added statements will be overwritten with newly generated BASIC code. We call this “leaving the model”

Entry screens keep all their custom code within the model.

Reports keep no custom code within the model.

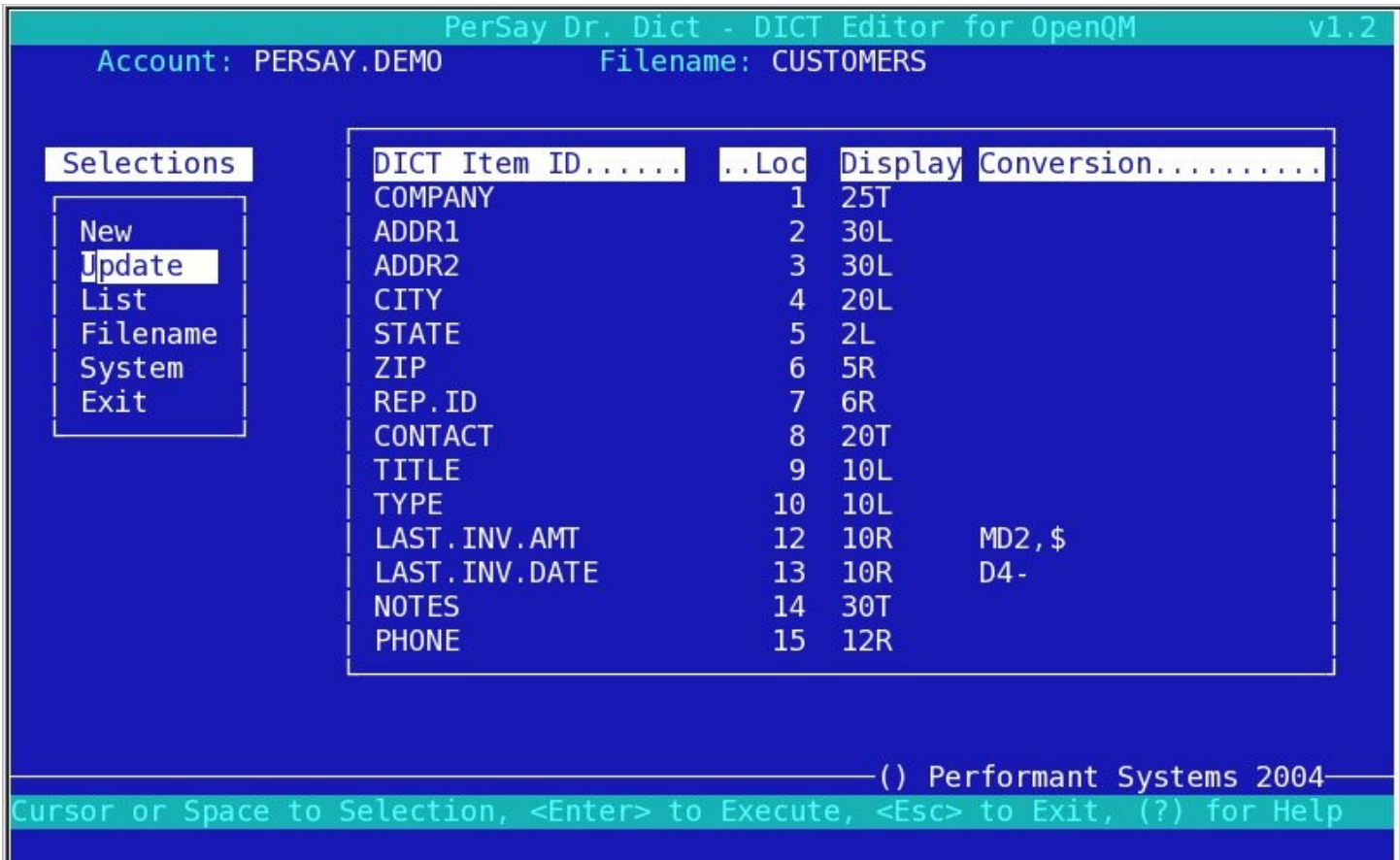
### ***About “Leaving the Model”***

While lots of features are built into PerSay programs, it is only a matter of time as to when a custom feature is needed that PerSay doesn't provide. On entry screens, you have options, you can call external subroutines, you can \$INCLUDE code sections as needed. More so on entry screens, you want to try to have all your custom code contained within the model. Recall that all model items, for entry screens and reports are kept in the PER.SCREENS file. Normally, you let PerSay made the contents of the PER.SCREENS item for you. You can certainly edit a PER.SCREENS item if you care to and are sure you know what you are changing.

If you do “leave the model” by entering custom code into the generated code, just be aware that should you take the “Generate” option for this model, there is no “Overwrite existing (Y/N)? “ prompt. PerSay always overwrites your existing code.

## Introduction to PerSay for OpenQM

### Dr. Dict – the PerSay DICTionary editor



Dr. Dict is available from many points of entry throughout PerSay. From the PerSay menu directly or from within the Entry Screen or Report layout editors. Also at a command line as the Verb DD.

Example:

```
:DD CUSTOMERS LAST.INV.DATE
```

where the Filename and DICT item name are optional.

### **Built-in Help**

With the highlighted pointer on any of the 6 Selections, hit (?) for a description of what that Selection does.

## Dr. Dict – DICT item properties

```
PerSay Dr. Dict - DICT Editor for OpenQM v1.2
Account: PERSAY.DEMO      Filename: CUSTOMERS

Selections
New
Update
List
Filename
System
Exit

(15 of 22)

DICT Item ID: LAST.INV.DATE

1) Type Code.....: D
2) Field Number/Exp.: 13
3) Conversion Code..: D4-
4) Display Heading..:
5) Display Format...: 10R
6) (S)ingle (M)ulti.: S
7) Association Name..:

8) Description
The date of the last invoice. Year is assumed to be current year.

-----() Performant Systems 2004-----
Change (1-8), (F)ile, (C)opy, (D)elete or <Esc> to exit: __
```

The building of DICT items is part of the art of design in a multivalue database system. Dr. Dict will accept most all the features of the rather complex range of features OpenQM offers for DICT items. It is beyond the scope of this Introduction to explain all of those features. See the OpenQM Query PDF file for full details.

### **Copy and Delete items**

We often have need to Copy and Delete DICT items. These are both done within the “Update” selection. Move the highlight pointer to an item of interest, hit <Enter> then (C)opy or (D)elete as needed.

# Introduction to PerSay for OpenQM

## Terminal Params Entry Screen

From the PerSay menu – select Terminal Params

```
PerSay - Terminal Parameters Entry v1.2
TERM.ID: linux      1) Name: Linux console
Editing Key Features -----Function Keys-----
2) Up.....: 27 91 65      9) F1: 27 91 91 65      14) F6: 27 91 49 55 126
3) Down...: 27 91 66      10) F2: 27 91 91 66      15) F7: 27 91 49 56 126
4) Right..: 27 91 67      11) F3: 27 91 91 67      16) F8: 27 91 49 57 126
5) Left...: 27 91 68      12) F4: 27 91 91 68      17) F9: 27 91 50 48 126
6) Insert.: 27 91 50 126   13) F5: 27 91 91 69      18) F10: 27 91 50 49 126
7) Delete.: 27 91 51 126
8) ShftTab: 2
Graphics Characters - Line Drawing
19) Top Int...: 119      24) Vert Line 1: 120      28) Top Right...: 107
20) Middle Int: 110      25) Vert Line 2: 120      29) Top Left....: 108
21) Bottom Int: 118      26) Horz Line 1: 113      30) Bottom Left.: 109
22) Left Int...: 116      27) Horz Line 2: 113      31) Bottom Right: 106
23) Right Int.: 117
Block Drawing
32) Full Density.: 97      35) Graphics On...: 27 40 48 39) Home...:27 91 49 126
33) Half Density.: 97      36) Graphics Off.: 27 40 66 40) End...:27 91 52 126
34) Light Density: 97      37) 80 Col. Mode.: 27 96 58 41) Pg Up...:27 91 53 126
38) 132 Col. Mode: 27 96 59 42) Pd Down:27 91 54 126
Change (1-42), (F)ile, (D)elete, (T)est Keys, (S)how Graphics or <Esc>: 
```

Here is a mind-numbing list of all of the ASCII codes that PerSay keeps in the PER.TERM item related to Term Types. As systems, emulations and keyboards vary, this is the manner PerSay keeps all these parameters together.

Use the (T)est Keys option to have PerSay show you what ASCII number codes each key sends out, often multiples. Taking note of these codes, you can enter them into the Terminal Parameters screen as needed. Most of the popular Term Types are already provided as items in the PER.TERM file. Go ahead and copy any PER.TERM items as you need to match your terminal environment.

The four arrow keys are crucial to operation of PerSay and PerSay generated screens. The backspace key is looking for ASCII 8 but will accept other values as shown above. PerSay Screen navigation is done by hitting either <Enter> or <Tab> to progress to the next field, with the option of <Shift><Tab> to regress a field. Often, keyboards are inconsistent about the <Shift><Tab> keystroke, so PerSay screens also accept <Ctrl>B for an equivalent to <Shift><Tab>.

Other keyboard equivalents PerSay accepts are: <Ctrl>D for Delete and <Ctrl>N to toggle Insert mode.

## Introduction to PerSay for OpenQM

### Define Default Parameters

Moving along in our tour of PerSay, select “Edit Defaults” from the PerSay menu for this screen.

```
PerSay - Define Default Parameters v1.2
ACCOUNT: PERSAY.DEMO

1) Editor Verb.....: SED
2) BASIC Programs File: BP
3) Entry Screen Style.: CHAR

Change (1-3), (F)ile or <Esc> to exit: █__
```

- 1) Editor Verb – What should PerSay execute to edit an Item? Usually, ED, SED or WED.
- 2) BASIC Programs File. Where should PerSay write BASIC program code into. PerSay will make a file for you if needed, see Create a File from the PerSay Menu.
- 3) Entry Screen Style – CHAR or LINE. The default CHAR (for character based) will create screens that work line PerSay itself does, with the full cursor based entries with Insert mode toggle, <Del> key, <Home>, <End> and <Esc> as universal exit key. The LINE based screens use simple INPUT statements and offer low functionality in today's technology.

## Introduction to PerSay for OpenQM

### Global File Updates – PER.GLOBAL

Select “Global Updates” from the PerSay menu.

This screen can also be run directly at a command line as RUN PER.BP PER.GLOBAL or just PER.GLOBAL

```
PerSay Global File Updates v1.2

1) Filename : CUSTOMERS

2) Selection: WITH LAST.INV.AMT GT "100.00"

3) Replace AMC   With Value.....
  1. 23          Y
  2. 24          (24)*10
  3.
  4.
  5.
  6.
  7.
  8.
  9.
 10.
 11.
 12.

Change (1-3), (S)tart or <Esc> to exit: 
```

Enter a Filename, a Selection and then up to ten different replacement values or formulas, enter (S) to Start and off we go. This is a convenience tool that comes in handy often.

You can either replace an attribute with a static value like the “Y” shown here or replace it with nothing which makes that attribute null.

Formulas can be entered like the one shown here which replaces attribute 24 with the contents of attribute 24 times 10.

Note that if a select list is present when PER.GLOBAL is run from a command line, that selection is still in place when PER.GLOBAL executes.

## Introduction to PerSay for OpenQM

TEST.BP items

There are a number of items in the TEST.BP file as a convenience for the developer to look at or mess around with.

```
TEST.BP.....  
$BASIC.OPTIONS  
LIST.ACCOUNTS  
TEST.COLORS  
TEST.LOCK  
TEST.PER.ENTER.MV  
TEST.PER.SCR.BOX  
TEST.PER.TEXT  
TEST.VIDEO  
8 records listed.
```

First note the \$BASIC.OPTIONS item. OpenQM has a clever and useful feature of having it's compiler look to this item for directives. Look at this default item and the OpenQM BASIC code PDF for further explanation of what goes in here. Most recommended to copy this item into your own BP file.

As PerSay makes extensive use of the PER.INC \$INCLUDE items, you can choose to make changes to the Color scheme used in PerSay or it's generated programs by making changes to the PER.CHAR.SETUP item. To figure out what colors you want, use the TEST.COLORS program shown above.

There are three programs here that demonstrate how to integrate 3 accessory features that come with PerSay, they are:

```
TEST.PER.ENTER.MV   - Provides a columnar list of multi values, mostly for Report front  
end                 screens, as an Entry screen already has multivalue entries.  
  
TEST.PER.SCR.BOX    - Just draws a graphic box from point A to point B.  
  
TEST.PER.TEXT       - integrated use of a line based text editor similar to Code Entry.
```

Feel free to try any or all of these programs to discover other features provided by PerSay tht you may find useful.

\*\*\*