SimpList
Version 1.3
By Steve Grant

Introduction

Readers of the February 2005 issue of this publication may remember a review of an ISPF productivity tool named SimpList, from MacKinney Systems. This product brings to the mainframe environment some of the same concepts and ease-of-use associated with working on a desktop system, shortening the learning curve for those new to the z/OS platform and dramatically improving the productivity of both rookie and veteran. It is packed with features, indeed too many to describe at length here. I leave that task to the detailed User Guide shipped with SimpList. Instead, this article will focus on the new functionality delivered in version 1.3, a beta release of which I recently had the opportunity to test.

More than just a Data Set organizer

A brief product overview may be appropriate to help put the new features in context. Although the marketing material contains the tagline "surf the mainframe"™, SimpList is not a GUI, but rather an ISPF dialog that installs on the mainframe, runs on the mainframe, and interfaces with just about everything else that runs on the mainframe (e.g. BookManager, DB2, HSM, SuperC, AMBLIST, third-party vendor products, etc). The power of SimpList comes from its ability to store the names of over 2,500 objects (i.e. not just datasets, but also DB2 tables, DB2 commands, BookManager bookshelves, TSO and ISPF commands and much more) in lists of 'favorites', and to use the same panel to easily launch functions (Edit, Copy and Print are just a few examples) against any object. For example, selecting a dataset for Edit opens the dataset in ISPF Edit, while selecting a DB2 table for Edit opens the DB2 table editor configured for your site.

Apart from the one-time keying of object names, all navigation and object selection can be achieved via one or two mouse clicks. All SimpList panels come with ample point-and-shoot support, hence the "surfing" reference. The object list has effectively become my 'home page' on the mainframe,

allowing me to focus on *what* I need to do, instead of the minutiae of *how* to do it.

SimpList indexes each of the 26 available object lists with the letters A through Z. A short alias of your choice can also be assigned to each list, and is displayed with the index value to remind you of each list's contents. For example, the objects on list 'C' could be given an index name of 'COBOL', and the objects on list 'J' could be given an index name of 'JCL' (etc).

All objects in a list are numbered (1 to 99), and if you choose not to point-and-shoot, you can select an object by entering its associated number or label (more on labels shortly) in the primary command area. Version 1.3 now gives you the option of specifying a member name or pattern as a command line argument after entering an object number or label. For example, I could enter '2 PGM*' to select the object on line 2 and see all the members that start with PGM.

SimpList or 3.4?

In addition to supporting multiple object types, SimpList replaces the standard ISPF member list and the DSLIST panel (ISPF option 3.4) with its own versions of these, considerably extending the functionality of each with primary commands and functions that work in *addition* to the regular member list and DSLIST functions. For example, function 'I' (Information) on SimpList's DSLIST not only displays dataset attributes but allows any of the displayed attributes to be changed simply by entering the new values. By performing one step on a single panel, SimpList can re-size a dataset in seconds. Accomplishing the same result using native ISPF involves several steps executed on different panels, and requires considerably more time.

The SimpList version of a member list is called the Advanced Member List (AML), and offers some great tools that merit special mention: the SEE command filters a member list based on matching member names and/or multiple member patterns and/or even statistics (e.g. size, creation date, etc.); SEEK filters the list based on the contents of members (e.g. a specified string is matched), and because its results are nested you can issue several

SEEKs in succession to filter the list down to those members containing 'x' *and* 'y' *and* 'z'.

New 'Label' feature

One of my favorite new features is the ability to give any type of object a label. A label is a user defined alias that begins with a period. For example, I could choose to assign ".JCL" as a label for a favorite JCL library. Not only is a label more meaningful (and easier to remember) than an object name or number, but it also makes the object accessible from any object list within SimpList. This was welcome news to my middle-aged memory ("Now, what list was that object on?").

Not happy with the labels you have created? Simply enter a lone period on the SimpList command line, and you're presented with a panel where you may add, delete and update label definitions.

From within SimpList, objects can be accessed for the default function by simply entering the label name on the command line. If you wish to open an object with a different function, the function can be supplied as well. For example, enter '.JCL V' to View whichever object is associated with the .JCL label.

One of the most convenient features of labels is that objects can be selected from *any* ISPF command line, even outside of a SimpList session. For example, I could enter "BR .JCL" on any command line to browse a JCL library, or "ED .TBL" to edit a DB2 table, or "VI .COB" to view a COBOL library (etc).

New 'Symbolics' feature

Developers who work in multiple test environments will want convenient access to multiple dataset objects that are *almost* identically named, such as 'TEST.ENV1.LOADLIB', 'TEST.ENV2.LOADLIB' and so on. The 'R' (Repeat) line command which is available on the object list panel makes it particularly easy to succumb to this temptation. Better to avoid this approach because it depletes the supply of unused entries in an object list. An alternative would be to use the built-in support for wild cards to generate a list of all data sets that match a pattern; e.g. 'TEST.ENV*.LOADLIB'. The problem with this second method is that after the

list of matching data sets has been displayed, you then have to tab down and select the data set you want.

With the new symbolics support, a single object containing 'TEST.ENV&NUM..LOADLIB' could be stored in a list. When selected, a pop-up window prompts for the value of &NUM to be entered. Simply type the number (e.g. '2'), and you're directly in the data set you want to be in.

Another excellent application for a symbolic is a DB2 table qualifier. For example, by creating an object containing '-&AUTHID..CUSTOMER_ACCOUNT', you can browse any instance of the table by changing the value of the AUTHID symbolic.

When creating a symbolic, you must specify it's longevity by choosing to be prompted for a value (a) every time the symbolic is referenced, or (b) only once per SimpList session, or (c) never again, saving that value forever. This latter option is the source of a minor gripe: SimpList provides no means to update or delete a symbolic value once option (c) is chosen. ISPF dialog developers will realize they can easily code a simple CLIST or Rexx program to VERASE the variable, but it would be nice if this ability was somehow 'built in'. (Update: SimpList version 2.0 now has this feature!)

TSO and ISPF commands can be stored as objects

Thanks to new support for commands as object types, there is relief for anyone who has struggled to recall the names or syntax of infrequently-used commands. Gone is the tedium of combing through manuals or online help, because any command that can be entered on an ISPF command line can now be stored as an object and executed with a mouse click. Remember that any object may contain symbolics, so you can use them strategically as arguments of saved commands to have SimpList prompt you for the values when you select the object. You can also use system symbolics so you don't have to be prompted. For example, TSO PROFILE PREFIX(&ZUSER) could be stored as a command, so simply selecting it would automatically set the TSO prefix to the value of the &ZUSER system variable.

New 'MRG' command

A powerful new MRG command has also been added to SimpList. It merges the directory information of up to four libraries (for display only - no physical updates happen) and displays the merged member list on an AML. The MRG command is entered on the command line of any object list, followed by up to 4 object numbers and an optional member name or pattern. Each number refers to a partitioned data set (PDS or PDSE format) on the current object list. For example, if the current object list contains the objects shown below, and 'MRG 1 2 3 4' is entered on the command line, the resulting merged member list is shown by the image in Figure 1.

```
_   OBJ   1   ?===> 'PROJECTA.TESTLIB1.COBOL.SOURCE'
_   OBJ   2   ?===> 'PROJECTA.TESTLIB2.COBOL.COPYBOOK'
_   OBJ   3   ?===> 'PROD.COBOL'
_   OBJ   4   ?===> 'PROD.COPYLIB'
```

MRG is a great gadget to have in your toolbox for several reasons, perhaps the best of which is the fact that any member whose name appears inside another member can be selected using point-and-shoot, even if the two members are in different libraries. For example, if a program in one library contains a reference to a copybook in another library, that copybook can be opened by simply entering a launch command (e.g. VI for View) and pointing the cursor at the copybook name.

Combining the new features yields numerous possibilities. One of my favorites is the ability to open, from *any* ISPF command line, a SimpList AML containing a merged member list from four different libraries. To do this I simply enter "BR .SOURCE" on any command line, where .SOURCE is a label I gave to a MRG command that merges my libraries. The performance of the MRG command is excellent. I tested a scenario with four datasets where the resulting merged list contained over 7,700 members, and the AML opened with sub-second response time.

Features for 'Power Users'

ISPF dialog developers from 'dabbler' to 'power user' will be interested to know that SimpList supplies an API (Application Programming Interface) and numerous 'hooks', allowing the integration of SimpList functions with their own

utilities. User Defined Functions (UDFs) can be employed to tailor SimpList to individual needs. As well, the AML provides the 'T' (TSO) and 'O' (macrO) functions, for invoking TSO commands and Edit macros of your choosing against some or all of the members in a list. A detailed, context-sensitive help facility ensures that answers to any questions are never far away.

Installation

SimpList installation is very easy and takes less than an hour. Anyone can install it, even if they only have authority to alter their own datasets, since *nothing* needs to be authorized, compiled, linked, added to the link-list or IPLed! The product ships with several *free* generic utilities that can be used independently of SimpList forever, even after the SimpList license or free trial has expired. If I ever have the misfortune to work at a site that doesn't

have SimpList, I'll be sure to take the free utilities with me. One example is SLX0001R, which is the initial macro that controls extended edit/view functions. For example, the point-and-shoot FIND commands (which can be used to find other occurrences of whichever string the cursor is pointed at) are controlled by the SLX0001R macro.

Conclusion

In my experience, SimpList offers such convenience, improved productivity and customization potential that I can not imagine working on a mainframe without it. There are too many features to even list, never mind describe, and its economical pricing (contact the vendor at www.mackinney.com for details) and short learning curve will yield an extremely early return on investment in shops of all sizes.

Figure 1 – SimpList Advanced Member List resulting from a MRG command

```
AML 1:    Merged Libs 1-4                          ROW 00001 OF 07725
COMMAND  ?==>                                       SCROLL ?==> CSR

CHGALL    CMDS      COMPRESS   CONFIRM   DOWN      END       EXHELP     Find       INFO
LEFT      Locate    MACSET     MLC       MRGSET    PRINTML   PRINTSET   PROFile    REFresh
RESet     RFIND     RIGHT      RSEEK     SAVE      SEE       SEEK       SEEKSET    Select
SORT      TSOSET    UP
FUNCTION ?==> B   Browse Copy Delete Edit Jobsub Move traNsfer macrO Print
                  Rename stAts Tso View Where eXclude 1,2,3=user defined

B  Name      Newname    Lib     Size     Created            Changed               ID
_  AAMOD01              1        27     1994/01/25   2000/03/08 11:07:58     USER01
_  AAMOD02              3         2     1997/12/12   2001/06/06 10:36:16     USER01
_  AAMOD03              2        22     1995/01/23   1995/06/21 07:50:39     USER01
_  AAMOD04              2        20     1995/03/10   1995/04/01 16:08:45     USER02
_  AAMOD05              2       134     2004/11/10   2004/11/10 15:13:36     USER01
_  AAMOD06              2        15     1998/07/29   1998/07/29 15:22:31     USER03
_  AAMOD07              2        43     1995/12/11   1995/12/11 11:30:35     USER04
                                                                              01
  This is a merged member selection list. Primary command MRGSET and line     01
  command W are applicable. To see which libraries are merged, point the      02
  cursor at the panel title. Press HELP for more information.                 01
                                                                              03
_  AAMOD13              1       712     2002/05/09   2002/05/09 14:01:15     USER01
```