

Contents

Screen Shots.....	2
Help Services.....	6
Source Code	8
Display DDS.....	17

Screen Shots

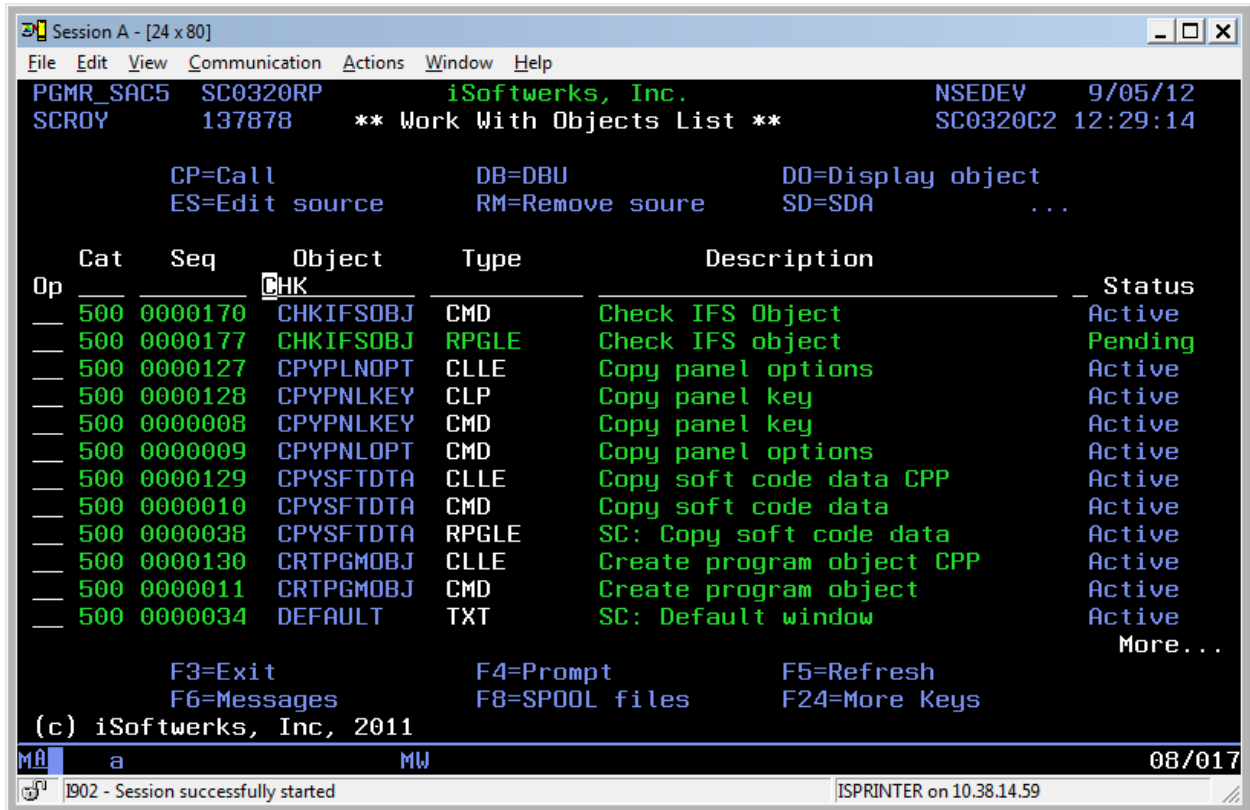


Fig. 1

This is a typical subfile display, to borrow from IBM's lexicon, CUA (Common User Access). The presentation is the same as IBM system panels, functions at the bottom, options at the top, data list in the middle.

SC0320RP Program Screen Captures

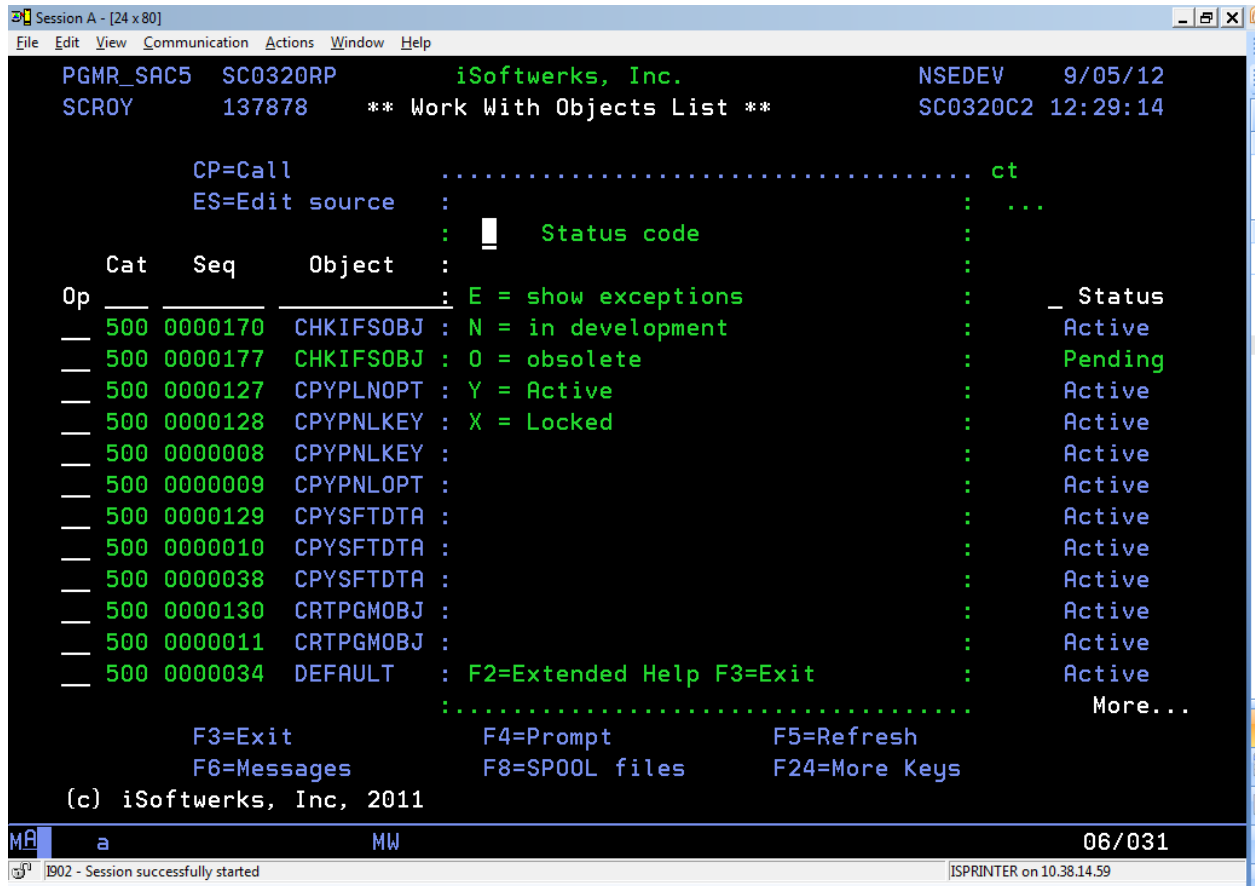


Fig. 2

The prompt process (Fig. 2) shows a window with a list of options and an input capable field which will return the value to the program. It is a text-based help function provided by the SoftCode service programs. It can be changed to appear in a different location, or a different size, simply by changing a file. (No program modification required.) The size of the input capable field may also be changed without re-compiling any objects.

SC0320RP Program Screen Captures

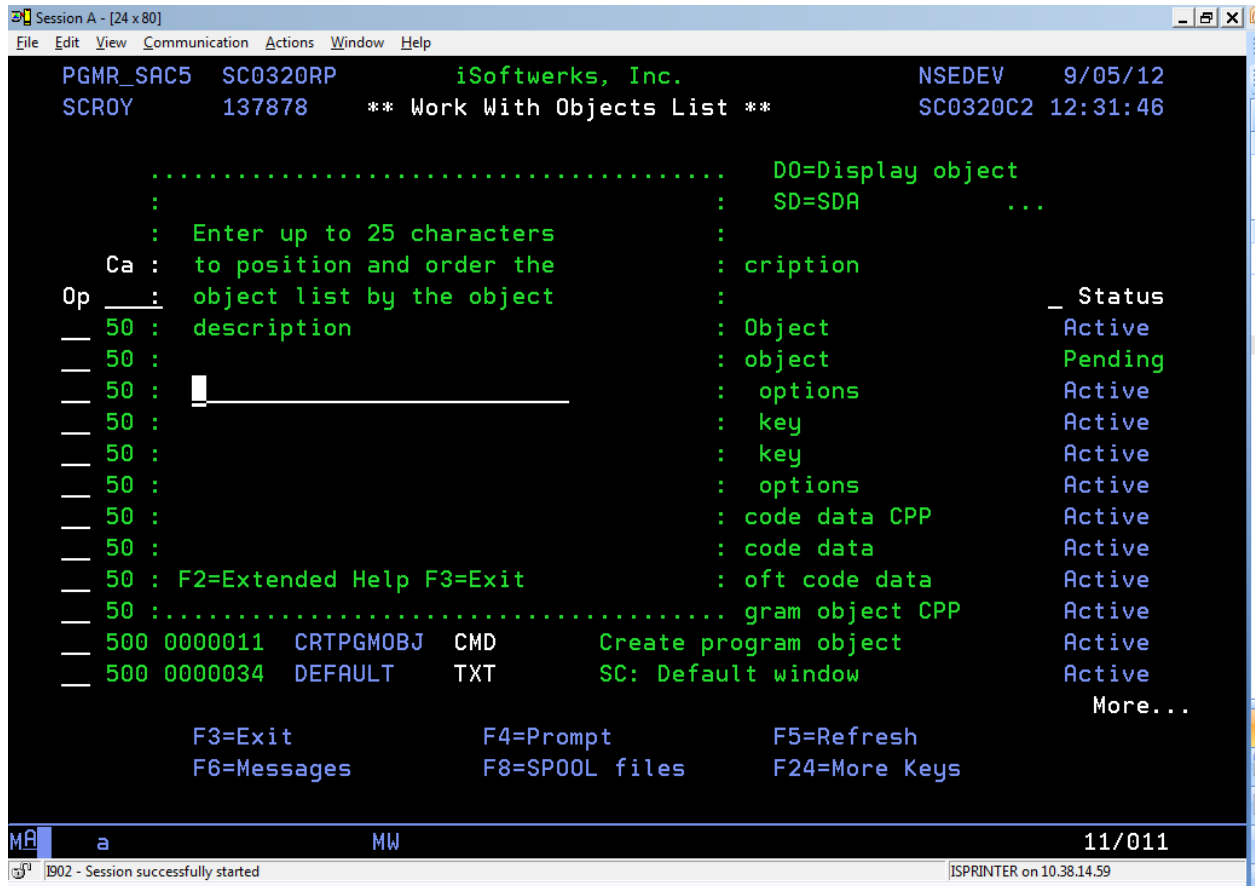


Fig. 3

In this screen capture the prompt was issued when the cursor was over the object description field. The data list may be sorted, or filtered by any of the fields over the columns. Entering up to 25 characters of the object description will sort the data list by object name, positioned to the first entry that matches, or is greater than the search characters entered. By the way, this prompt screen is the same display and program that you see in Fig. 2. Even though the window is in a different location, and a different size, and contains a different-sized field for input, it is the same display and the same program. In both Fig. 2 and Fig. 3, the prompter is simply painting a 1920-byte data stream over the existing panel.

SC0320RP Program Screen Captures

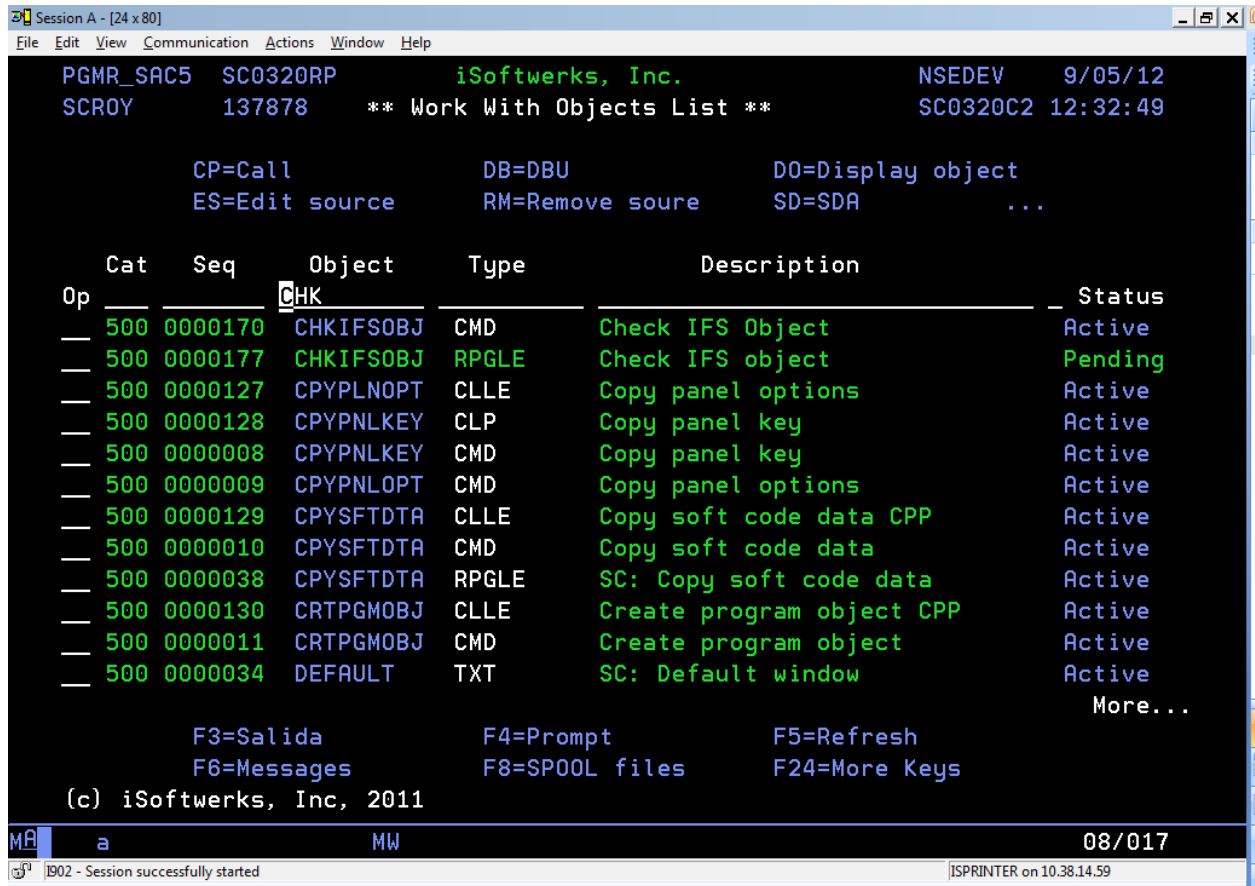


Fig. 4

There is not much to see in Fig. 4—except the text for F3 is not Salida, not Exit. Without changing a display, or re-compiling a program the SoftCode function key editor (EDTPNLKEY) has been used to change the presentation text from English to Spanish (Salida = Exit). Any text, function key, or program option may be changed without any program changes or display file changes. The process takes a few seconds.

Help Services

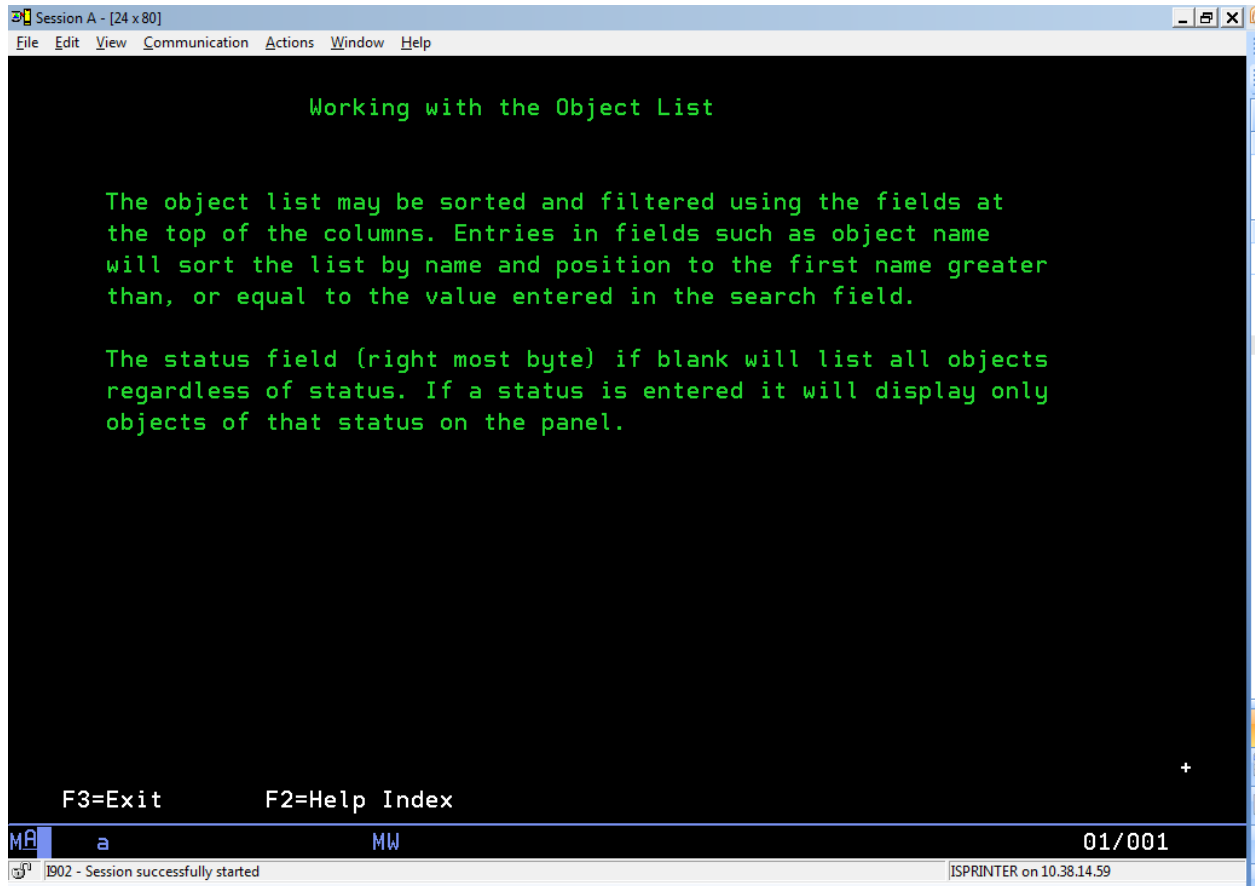


Fig. 5

There is program help text available as well as the prompter. The service programs actually provide two separate help processes. This (Fig.5) is an example of the 5250HELP service which provides a simple index help facility. The text can be as simple as a few lines to describe a function or as complex as a complete text description of the entire application.

SC0320RP Program Screen Captures

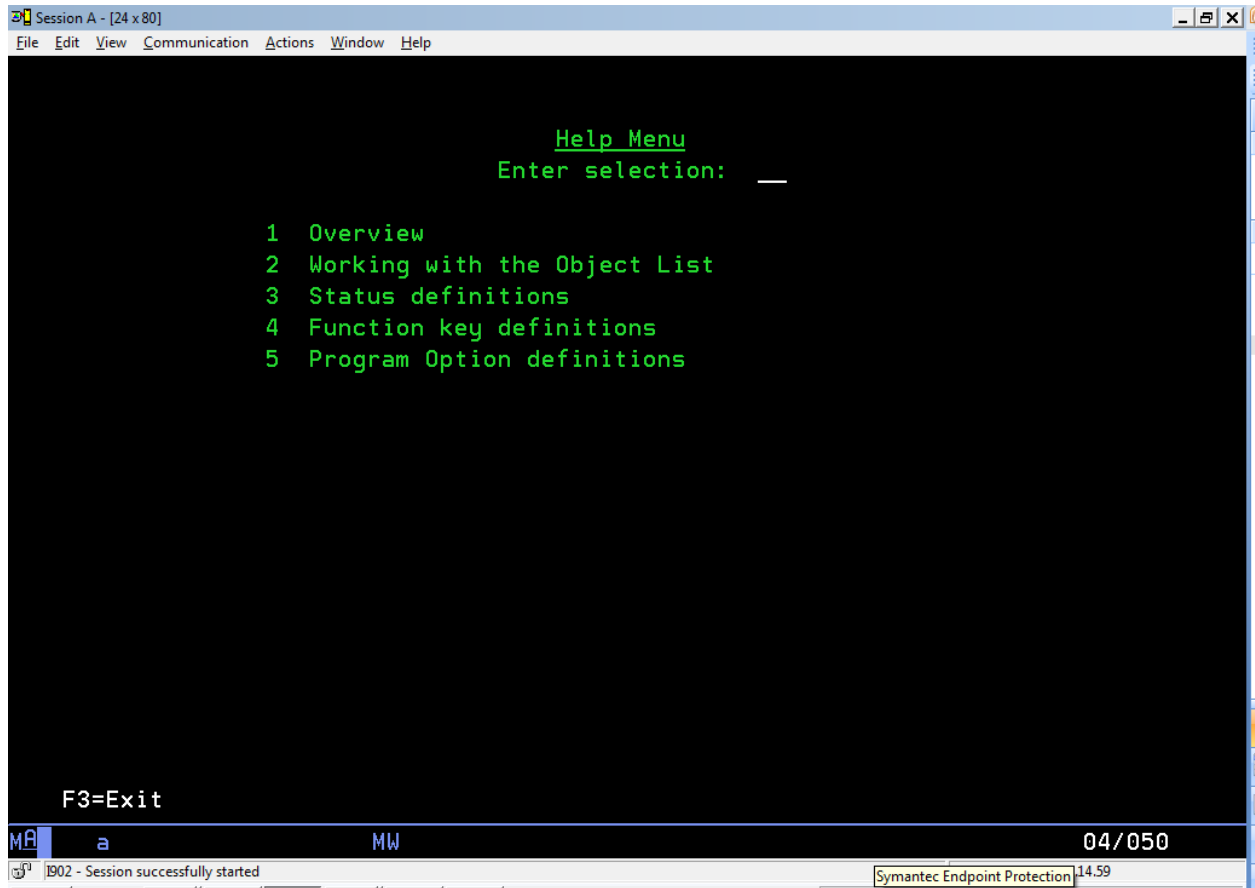


Fig. 6

This screen capture (Fig. 6) is a quick look at the index feature of the 5250HELP service. Though it is invoked from SC0320RP, it is a part of the services and not a part of the application program itself. Therefore, it can be maintained independently from the application program.

A completely separate module of 5250 help may be used by the application. If there is a URL that opens a PDF, Word document, or HTML page, the help services will issue a command through the system PC Organizer and launch a browser to display the help document. It is relatively simple, and if an intra-company server has been set with a directory for such documents, it is fast and easy to create browser-based application help.

SC0320RP Program Screen Captures

Source Code

```

/TITLE ** Work With Softcode Entries *
H DEBUG(*YES)
H OPTION(*SRCSTMT : *NODEBUGIO) DFTACTGRP(*NO) ACTGRP('QILE')
H BNDDIR('QC2LE':'SC0000_BD')
*****
* Program Name - SC0320RP
*
* Function      - This program was designed to allow a user to
*                work with expansion project objects
*
* Programmer    - Steve Croy                09/03/09
*****
* Compile instructions
*CRTPGM *RPGLE
*DBGVIEW *SOURCE
*
*****
*                Modification log
*
* Date      Programmer      Description
*
*****
FSC0320DF  CF  E                WORKSTN
F                                SFILE(SC0320S1:RRNSI)
F                                INFDS(DSPDS)
*=====
D FUNCTIONKEY  E DS                EXTNAME(SCKEYSPF) qualified      Function keys
D PGMDS        E SDS                EXTNAME(SCPSTSPF)                Pgm status map
D DSPDS        E DS                EXTNAME(SCDSPFPF)                Display INFDS
D MACDS        E DS                EXTNAME(SCFUNCPF) INZ            Key map
D OPTDS        E DS                EXTNAME(SCOPTNPF) INZ            Option map

D OBJECT       E DS                extname(SCOBJSPF)
D before       E DS                extname(SCOBJSPF) qualified
D after        E DS                extname(SCOBJSPF) qualified
D Macro        E DS                extname(SCMACRPF)
D Function     30a                  overlay(Macro:1)

D FormatSQL     PR                    ExtPgm('SC0325RP')
D pFcat        3 0
D pFseq        7 0
D pFobj        10a
D pFtyp        10a
D pFdsc        30a
D pFsts        1a
D pfSel        256
D pfOrd        256

D FilterSelect PR                    ExtPgm('SC0330RP')
D pFcat        3 0
D pFseq        7 0
D pFobj        10a
D pFtyp        10a
D pFdsc        30a
D pFsts        1a
D pfACT        3          CONST

D WithParms1   PR                    extpgm(SUBPGM)
D parm01       10
D parm02       10
D mode         1
D return       7

D CommandLine  PR                    extpgm('QUSCMDLN')

D WorkSource   PR                    extpgm(SUBPGM)

```


SC0320RP Program Screen Captures

```

D parm01          10
D parm02          10

D WorkObject      PR          extpgm(SUBPGM)
D parm01          10
D parm02          10

D NoParms         PR          extpgm(SUBPGM)

D DBUfile         PR          extpgm('SC0338CL')
D parm01          10
D parm02          10

*-- Prototype Copybooks
/copy qrpglesrc,SC0000_pr
*---
d indPtr          s          *   inz( %addr(*in) )

* define named indicators
d indicators      ds          99   based( indPtr )
d ScreenChange   n          overlay( indicators : 22 )
d SflControl     n          overlay( indicators : 50 )
d SflDisplay     n          overlay( indicators : 51 )
d SflInitialize  n          overlay( indicators : 52 )
d SflClear       n          overlay( indicators : 53 )
d SflEnd         n          overlay( indicators : 54 )
d SflDelete      n          overlay( indicators : 55 )
d SflNxtChange   n          overlay( indicators : 58 )
d SflMSGQdisplay...
d                n          overlay( indicators : 59 )

D Quit           pr
D Exit           pr          extproc('exit')
D                3u 0 value

* Default cursor position
D #DFPOS         DS          INZ
D #DFROW         2 0 INZ(8)
D #DFCOL         3 0 INZ(17)
*-----
* Define constants
*-----
D #TITLE         C          CONST('** Work With Objects List **')
D #MSGF          C          CONST('SCMSSGF')
D #SAME          C          CONST('*SAME')
*-----
* START of work fields
*-----
D auth1          S          3     inz('999')
D CATEG          S          3S 0  inz(500)
D cl             s          3s 0
D CMDKEY         S          720
D dspatr         S          1
D EOFIND         S          1     INZ('0')
D ErrorOccurred...
D                S          n
D FKEYYDS        S          1
D fmt            S          10    inz('BEGIN')
D INDLR          S          1
D Keyidx         S          2
D listAction     S          n
D MessageToDisplay...
D                S          n
D MoreRecordsRemain...
D                S          n    inz(*OFF)
D M              S          3S 0
D MorOpt         S          1
D MSG            S          80
D MSGDTA        S          132
D MSGF           S          10
D MSGPGM        S          10

```

SC0320RP Program Screen Captures

```

D MSGRLQ          S          5
D NoMoreRecords...
D                S          n
D O              S         3S 0
D objectFound    S          n
D OptIdx         S          2
D OPTION         S         720
d orderBy        s         256a
d P$ERR          s          7
d P$MOD          s          1
D p$mode         S          1
D parm01         s          10
D parm02         s          10
D parm03         s          10
D parm04         s          10
D parm05         s          10
D parm06         s          1
D parm07         s          1
D parm08         s          1
D parm09         s          1
D parm10         s          10
D PASSC         S          1
D PnlNam        S         10a   inz('SC0320C1')
D pmt           S          10   inz('PROMPT')
D promptDta     S         256
D RCDNBR        S          5 0  INZ(1)
D RRNSI         S          4 0
D RW            s          3s 0
D SAVRRN        S          4 0
D ScanRequested...
D              S          n   inz(*ON)
d selectOnly    s         256a
D SFLLOD        S          4 0
D SFLMAX        S          4 0  INZ(12)
D SFLPOS        S          4 0
D SFLRCN        S          4 0
D thisOption    S          10
D thisFormat    S          10
D thisPgm       S          10
D ttypatr       S          1
D UsrNam        S         10a
D Work8         S          8
D Work6         S          6
*-----
* END of work fields
*-----
/free
  CheckAuthority(USER:PRGNAM:CATEG:PASSC:AUTHL);
  IF PASSC <> 'P';
    QUIT();
  ENDIF;
  z$seq1 = #TITLE;
  functionKey = FunctionKeys();
  MSGID = 'MIS0001';
  EXSR @GetMsg;
  FilterSelect ( zfcatt: zfseq: zfoj: zftyp: zfdsc: zfst: 'GET');
  EXSR @reset;

  DOU keyPressed = functionKey.F3;

  IF MessageToDisplay;
    SflMSGQdisplay = *ON;
  ELSE;
    SflMSGQdisplay = *OFF;
  ENDIF;
  // write subfile message queue

  WRITE SC0320C2;
  IF ScanRequested;
    #ROW = #DFROW;

```

SC0320RP Program Screen Captures

```

#COL = #DFCOL;
ENDIF;

// Display subfile, test for EOJ, and function requested.

rrnsi = 0;
sflControl = *ON;
IF sflrcn > 0;
    sflDisplay = *ON;
ENDIF;
WRITE SC032001;
EXFMT SC0320C1;
sflControl = *OFF;
sflDisplay = *OFF;
getCsrLoc(ROW:COL:rw:cl);
#ROW = rw;
#COL = cl;

IF MessageToDisplay;
    RmvMessage(prgnam);
    MessageToDisplay = *OFF;
ENDIF;

ThisFormat = fmtnam;
ThisPgm = prgnam;
GetFunction(thisPgm:thisFormat:keypressed:fkeyid:macro:authl);

SELECT;
WHEN KeyPressed = functionKey.ENTER;
    EXSR @ENTER;
WHEN KeyPressed = functionKey.ROLLUP;
    EXSR @LOAD;
WHEN KeyPressed = functionKey.ROLLDN;
    EXSR @DOWN;
WHEN KeyPressed = functionKey.F23;
    DisplayOptions(option: z$opt1: z$opt2: 0);
WHEN KeyPressed = functionKey.F24;
    DisplayKeys(cmdkey: z$key1: z$key2: M);
WHEN Function = 'EXIT';
    QUIT();
WHEN Function = 'CANCEL';
    EXSR @return;
WHEN Function = 'RESET';
    EXSR @RESET;
WHEN Function = 'PROMPT';
    EXSR @PROMPT;
    IF promptDta <> *blanks;
        PromptDta = *blanks;
        EXSR @RESET;
    ENDIF;
WHEN Function = 'HELP';
    HelpText(ThisPgm:fmt);
WHEN SUBOP = 'CALL';
    EXSR @CALLS;
WHEN Function = 'CMDLINE';
    CommandLine();
ENDSL;
CLEAR macro;
ENDDO;

/*=====
/* Process ENTER key
/*=====
BEGSR @PROMPT;
READ SC032001;
pmt = rtnfld;
Prompter(Thispgm:pmt:PromptDta);
SELECT;
    WHEN rtnfld = 'ZFSTS';
        zfsts = %trim(PromptDta);
    WHEN rtnfld = 'ZFSEQ';

```

SC0320RP Program Screen Captures

```

        promptDta = %xlate(' ':0':promptDta);
        zfseq = %dec(%subst(promptDta:1:7):7:0);
WHEN rtnfld = 'ZFCAT';
        promptDta = %xlate(' ':0':promptDta);
        zfcats = %dec(%subst(promptDta:1:3):3:0);
WHEN rtnfld = 'ZFOBJ';
        zfobj = %trim(promptDta);
WHEN rtnfld = 'ZFTYP';
        zftyp = %trim(promptDta);
WHEN rtnfld = 'ZFDSC';
        zfdsc = %trim(promptDta);
OTHER;
        PromptDta = *blanks;
ENDSL;
ENDSR;
/*=====
/* Process ENTER key
/*=====
BEGR @ENTER;
// Read and Update subfile records ...
IF Z$RRN1 > 0;
    DOU %eof(SC0320DF);
        READC SC0320S1;
        listAction = *off;
        IF NOT %eof(SC0320DF);
            IF Z$OPT <> *BLANK;
                listAction = *ON;
                thisOption = %triml(z$opt);
                ThisFormat = fmtnam;
                ThisPgm = prgnam;
                GetOption(Thispgm:ThisFormat:thisOption:macro:authl);
                SELECT;
                WHEN SUBOP = 'CALL';
                    EXSR @CALLS;
                WHEN function = 'PROCESS';
                    objectFound = retrieveObject( exobnm:exobtp );
                    IF objectFound;
                        objectFound = ObjectProcess();
                    ENDIF;
                WHEN function = 'RESET';
                    objectFound = retrieveObject( exobnm:exobtp );
                    IF objectFound;
                        objectFound = ResetProcess();
                    ENDIF;
                WHEN function = 'EXCEPTION';
                    objectFound = retrieveObject( exobnm:exobtp );
                    IF objectFound;
                        objectFound = ProcException();
                    ENDIF;
                WHEN function = 'OBSOLETE';
                    objectFound = retrieveObject( exobnm:exobtp );
                    IF objectFound;
                        objectFound = ObjectObsolete();
                    ENDIF;
                WHEN function = 'EXTEND';
                    objectFound = retrieveObject( exobnm:exobtp );
                    IF objectFound;
                        objectFound = ObjectExpanded();
                    ENDIF;
            ENDSL;
            z$opt = *BLANK;
            IF before <> after;
                SELECT;
                WHEN p$mode = 'C';
                    OBJECT = after;
                    exdesc = '*changed';
                WHEN p$mode = 'D';
                    exdesc = '*deleted';
                    *IN30 = *ON;
                ENDSL;
            ENDIF;

```

SC0320RP Program Screen Captures

```

                z$rrn2 = z$rrn1;
                UPDATE SC0320S1;
                *IN30 = *OFF;
                CLEAR MACRO;
            ENDIF;
        ENDIF;
    ENDDO;
ENDIF;
EXSR @READ;
ENDSR;
/*=====
/* Read the scan record format and move the scan value to the
/* appropriate field.
/*=====
BEGSR @READ;
    ScanRequested = *OFF;
    READ SC032001;
    IF ScreenChange;                                // Change indicator on
        ScanRequested = *ON;
        EXSR @RESET;
    ENDIF;
ENDSR;
/*=====
/* Initialize subfile, and reposition file for subfile load
/*=====
BEGSR @reset;
    rrsni = 1;
    rcdnbr = 1;
    sflrcn = 0;
    sflpos = 0;
    SflInitialize = *ON;
    WRITE SC0320C1;
    SflInitialize = *OFF;
    SflEnd = *OFF;
    ThisPgm = prgnam;
    ThisFormat = fmtnam;
    m = 0;
    o = 0;
    usrn = user;

    GetKeyText(thisPgm:thisFormat:cmdkey:auth1);
    GetOptText(thisPgm:thisFormat: option: auth1);
    DisplayKeys(cmdkey: z$key1: z$key2: M);
    DisplayOptions(option: z$opt1: z$opt2: 0);

// Format the SQL request ...
FormatSQL ( zfcats: zfsseq: zfoobj: zftyp: zfdsc: zfst:
           SelectOnly: orderBy);
CloseObjectCursor();
ClearObject();
SetObjectCursor(orderby: selectOnly);
FilterSelect ( zfcats: zfsseq: zfoobj: zftyp: zfdsc: zfst: 'SET');
EXSR @LOAD;
ENDSR;
/*=====
/* Subroutine to process rolldown key
/*=====
BEGSR @DOWN;
    Z$RRN2 = Z$rrn2 - SFLMAX;
    SFLPOS = (SFLPOS - SFLMAX);
    IF SFLPOS < 1;
        SFLPOS = sflmax;
    ENDIF;
    IF Z$RRN2 < 1;
        Z$RRN2 = 1;
        MSGID = 'MIS0006';
        EXSR @GetMsg;
    ENDIF;
ENDSR;
/*=====
/* Set lower limits using search argument, load subfile from DBF

```

SC0320RP Program Screen Captures

```

/* read file until end of file, or max nbr of records loaded
/*=====
BEGSR @LOAD;
  sfllod = 0;
  savrrn = z$rrn2;
  sflend = *OFF;
  NoMoreRecords = *OFF;
  DOU NoMoreRecords or sfllod >= sflmax;
    MoreRecordsRemain = NextObject();
    IF MoreRecordsRemain;
      OBJECT = GetObjectData();
      sflrcn = sflrcn + 1;
      sfllod = sfllod + 1;
      RRNSI = SFRLCN;
      Z$RRN1 =SFRLCN;
      z$opt = *BLANKS;
      z$desc = exdesc;
      zfstat = *blanks;
      SELECT;
      WHEN exproc = 'Y';
        dspatr = SetColor('BLU');
        typatr = SetColor('WHT');
        zfstat = dspatr + 'Active';
      WHEN exproc = 'E';
        dspatr = SetColor('YLW');
        typatr = SetColor('WHT');
        zfstat = dspatr + 'Review';
      WHEN exproc = 'X';
        dspatr = SetColor('GRN');
        typatr = SetColor('GRN');
        zfstat = dspatr + 'eXtended';
      WHEN exproc = 'N';
        dspatr = SetColor('GRN');
        typatr = SetColor('GRN');
        zfstat = dspatr + 'Pending';
      WHEN exproc = 'O';
        dspatr = SetColor('RED');
        typatr = SetColor('RED');
        zfstat = dspatr + 'Obsolete';
      OTHER;
        dspatr = SetColor('GRN');
        typatr = SetColor('GRN');
        zfstat = dspatr + 'Undefined';
      ENDSL;
      zfobnm = dspatr + exobnm + typatr + exobtp;
      WRITE SC0320S1;
    ELSE ;
      Sflend = *ON;
      NoMoreRecords = *ON;
    ENDIF;
  ENDDO;

  sflpos = sflpos + sflmax;
  z$rrn2 = (sflpos - sflmax) + 1;
  IF z$rrn2 > sflrcn;
    sflpos = sflpos - sflmax;
    z$rrn2 = savrrn;
    msgid = 'MIS0007';
    EXSR @GetMsg;
  ENDIF;
ENDSR;
/*=====
/* This subroutine allows program calls using pre-defined PLISTS
/*=====
BEGSR @CALLS;
  EXSR @SETPM;
  MONITOR;
  SELECT;
    WHEN CALLPM = 'WORKSOURCE';
      CALLP WorkSource(exombr: exobsr);
    WHEN CALLPM = 'WORKOBJECT';

```

SC0320RP Program Screen Captures

```

        CALLP WorkObject(parm01: parm02);
    WHEN CALLPM = 'PLIST1';
        CALLP WithParms1(parm01: parm02: p$mode: p$err);
    WHEN CALLPM = 'DBUFILE';
        CALLP DBUfile(exobnm: exoblb);
    OTHER;
        CALLP NoParms();
    ENDSL;
    ON-ERROR;
        P$ERR = 'MIS0012';
        EXSR @GetMsg;
    ENDMON;
    EXSR @RETPM;
ENDSR;
/*=====
/* This subroutine sets values for pre-defined PLISTS
/*=====
BEGSR @SETPM;
    parm01 = exobnm;
    parm02 = exobtp;
    parm04 = *blanks;
    p$err = *blanks;
    p$mode = subact;
    IF p$mode = 'A';
        parm01 = *blanks;
        parm02 = *blanks;
    ENDIF;
    IF listAction;
        IF RetrieveObject(exobnm:exobtp);
            before = GetObjectData();
        ELSE;
            CLEAR before;
        ENDIF;
    ENDIF;
ENDSR;
/*=====
/* This subroutine determines actions based on returned parms
/*=====
BEGSR @retpm;
    IF listAction;
        IF RetrieveObject(exobnm:exobtp);
            after = GetObjectData();
        ELSE;
            CLEAR after;
        ENDIF;
    ENDIF;
    IF p$err<>*BLANKS;
        msgid = p$err;
        MessageToDisplay = *ON;
        EXSR @GetMsg;
    ENDIF;
ENDSR;
/*=====
/* Get message text from message file and turn on message flag
/*=====
BEGSR @getmsg;
    msgdta = *BLANKS;
    msg = *BLANKS;
    msgf = #MSGF;
    MONITOR;
        RtvMessage(msgid:msgf:msgdta:msg);
    ON-ERROR;
        MessageToDisplay = *ON;
    ENDMON;
    msgtxt = msg;
    EXSR @SendMessage;
ENDSR;
/*=====
/* Subroutine to send messages to program message queue
/*=====
BEGSR @SendMessage;

```

SC0320RP Program Screen Captures

```
msgdta = msgtxt;
msgpgm = PRGNAM;
msgrlq = #SAME;
msgf = #MSGF;
SendMessage(msgid:msgf:msgdta:msgrlq:msgpgm);
MessageToDisplay = *ON;
ENDSR;
/*=====
BEGSR @return;
  ErrorOccurred = CloseObjectCursor();
  *inlr = *on;
  return;
ENDSR;

/end-free
P Quit          b
/free
  ErrorOccurred = CloseObjectCursor();
  *inlr = *on          ;
  exit(0)              ;
/end-free
P Quit          e
```

The program is typical of development using SoftCode services. The entire program is 587 lines of source code (which includes comments). The program depends on various service programs, for help text, prompting, and for command execution. The function keys and the program options are not defined in the program, but are actually stored in a database file.

The display file is not much more complex. All function keys are defined to the display, and one display per program is the general rule, (with the exception of the subfile message queue). The display does not have any text referencing function keys, or program options, though there are fields defined to hold key values and option text. The fields are eventually populated by the SoftCode service program, depending on the program, panel name, and the authorization level of the user.*

Notice the application makes a distinction between QUIT and RETURN. In ILE setting on LR and returning does not affect the activation group per se. You may return and leave cursors open, or you may wish to terminate the activation group and allow the system to recover the resources. Hence the difference—RETURN will leave the activation group resources available to the next application. QUIT will terminate the program and recover the resources to the last activation group boundary.

* The SoftCode function and option editors do have a Function Authorization Level (FAL) built in, though it may or may not be used. The editor will not return text for any function or option a user is not permitted to view. And the editor will not execute a function or perform an option that a user is not authorized to perform.

SC0320RP Program Screen Captures

Display DDS

```

A*****
A* DISPLAY NAME - SC0320DF *
A* *
A* FUNCTION - This display is a list inquiry over the softcode *
A* object list *
A* *
A* PROGRAMMER - STEVE CROY 09/03/09 *
A*****
A*****
A* MODIFICATION LOG *
A* *
A* DATE PROGRAMMER DESCRIPTION *
A* *
A*****
A*%EC
A DSPSIZ(24 80 *DS3)
A PRINT
A INDARA
A CF01
A CF02
A CA03
A CF04
A CF05
A CF06
A CF07
A CF08
A CF09
A CF10
A CF11
A CF12
A CF13
A CF14
A CF15
A CF16
A CF17
A CF18
A CF19
A CF20
A CF21
A CF22
A CF23
A CF24
A ROLLUP(90)
A ROLLDOWN(91)
A HOME
A HELP
A R SC032001
A*%TS SD 20100331 131540 SCROY REL-V5R4M0 5722-WDS
A CHANGE(22)
A BLINK
A OVERLAY
A JOBNAM 10A O 1 2COLOR(BLU)
A PRGNAM 10A O 1 13COLOR(BLU)
A 1 29'iSoftwerks, Inc.'
A 1 61SYSNAME
A COLOR(BLU)
A 1 70DATE
A EDTCDE(Y)
A COLOR(BLU)
A USER 10A O 2 2COLOR(BLU)
A JOBNBR 6A O 2 13COLOR(BLU)
A Z$SEQ1 30A O 2 23DSPATR(HI)
A FMTNAM 8A O 2 61COLOR(BLU)
A 2 70TIME
A COLOR(BLU)
A Z$OPT1 60 O 4 11COLOR(BLU)
A Z$OPT2 60A O 5 11COLOR(BLU)
A 7 2' Cat Seq Object Type -

```

SC0320RP Program Screen Captures

```

A                                     Description
A                                     '
A                                     DSPATR(HI)
A                                     8 2'Op'
A                                     DSPATR(HI)
A      ZFCAT          3Y 0B 8 5DSPATR(HI)
A                                     EDTCDE(Z)
A      ZFSEQ          7Y 0B 8 9DSPATR(HI)
A                                     EDTCDE(Z)
A      ZFOBJ          10A B 8 17DSPATR(HI)
A      ZFTYP          10A B 8 28DSPATR(HI)
A      ZFDSC          30A B 8 39DSPATR(HI)
A      ZFSTS          1A B 8 70DSPATR(HI)
A                                     8 72'Status'
A                                     DSPATR(HI)
A                                     SFL
A      R SC0320S1
A*%TS SD 20091130 125427 SCROY      REL-V5R4M0 5722-WDS
A 58                                     SFLNXTCHG
A      EXOMBR        R      H      REFFLD(RSCOBS/EXOMBR *LIBL/SCOBJSP-
A                                     F)
A      EXOBSR        R      H      REFFLD(RSCOBS/EXOBSR *LIBL/SCOBJSP-
A                                     F)
A      EXOBNM        R      H      REFFLD(RSCOBS/EXOBNM *LIBL/SCOBJSP-
A                                     F)
A      EXOBTP        R      H      REFFLD(RSCOBS/EXOBTP *LIBL/SCOBJSP-
A                                     F)
A      EXOBLB        R      H      REFFLD(RSCOBS/EXOBLB *LIBL/SCOBJSP-
A                                     F)
A      Z$RRN1        4S 0H
A      Z$OPT          2A B 9 2DSPATR(HI)
A 30                                     DSPATR(PR)
A      EXCATS        R      O 9 5REFFLD(RSCOBS/EXCATS *LIBL/SCOBJSP-
A                                     F)
A      EXCSEQ        R      O 9 9REFFLD(RSCOBS/EXCSEQ *LIBL/SCOBJSP-
A                                     F)
A      ZFOBNM        21A O 9 17
A      Z$DESC        30A O 9 39
A      ZFSTAT        10A O 9 70
A      R SC0320C1
A*%TS SD 20100331 131540 SCROY      REL-V5R4M0 5722-WDS
A                                     SFLCTL(SC0320S1)
A                                     SFLSIZ(0013)
A                                     SFLPAG(0012)
A                                     BLINK
A                                     OVERLAY
A 51                                     SFLDSP
A 50                                     SFLDSPCTL
A 53                                     SFLINZ
A 55                                     SFLDLT
A 52                                     SFLCLR
A 54                                     SFLEND(*MORE)
A                                     CSRLOC(#ROW      #COL)
A      #ROW          3S 0H
A      #COL          3S 0H
A      Z$RRN2        4S 0H      SFLRCDNBR(CURS)
A      Z$KEY1        60A O 22 11COLOR(BLU)
A      Z$KEY2        60A O 23 11COLOR(BLU)
A*-----
A* SUBFILE MESSAGE QUEUE
A*-----
A      R SC0320S2
A                                     SFL
A      MSGKEY        SFLMSGRC(24)
A      PRGNAM        SFLMSGKEY
A      R SC0320C2
A      SFLCTL(SC0320S2)
A      SFLSIZ(2) SFLPAG(1)
A 59                                     SFLEND
A 59                                     SFLDSP
A                                     SFLDSPCTL
A N59                                     SFLCLR
A 59                                     SFLINZ
A      PRGNAM        SFLPGMQ

```