NAVAN LTD.

NUTIL

Navan Utilities for the iSeries

An overview

January, 2023

COPYRIGHT

Information in this document is subject to change without notice. The latest version of this manual can always be downloaded from the Navan Internet Website, which can be found at

http://www.navan.co.uk

VERSION 7 RELEASE 6, MODIFICATION 5

This edition applies to Version 7 Release 6, Modification level 5 of the NUTIL Navan Utilities Program Product.

Unless otherwise stated, all pages in this edition are to Version 7, Release 6 and refer to the iSeries Operating System i5/OS, Version 7.3 or greater.

Any references in this manual to Navan Utilities or NUTIL refer to standard Navan Utility products.

Information on these products can be obtained from your nearest Navan agent.

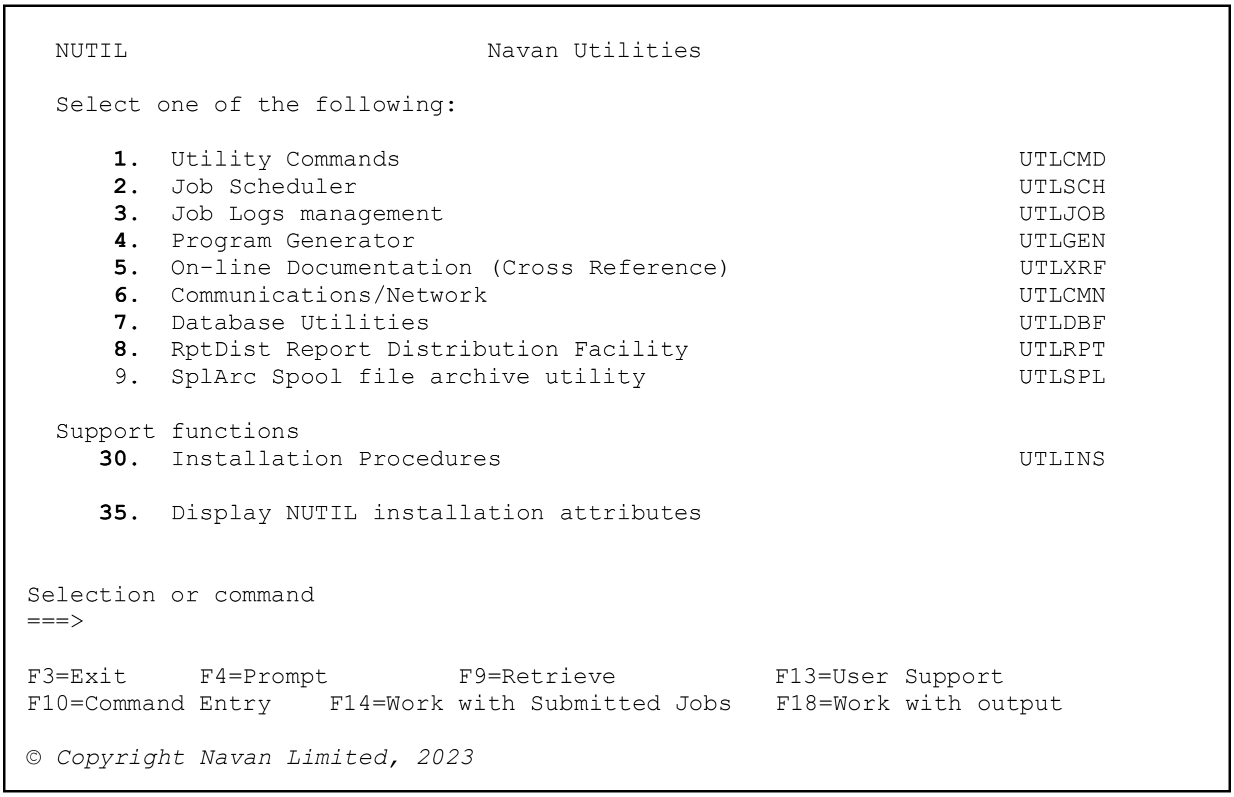
NUTIL - An overview

NUTIL (the Navan Utilities) is a diverse set of Programming and Operations utilities to complement and improve upon the functions available on your iSeries.

The NUTIL Main Menu

All major functions within NUTIL can be accessed via the NUTIL main menu.

```
===> GO NUTIL/NUTIL
```



```
  NUTIL                         Navan Utilities

  Select one of the following:

      1.  Utility Commands                                         UTLCMD
      2.  Job Scheduler                                            UTLSCH
      3.  Job Logs management                                      UTLJOB
      4.  Program Generator                                        UTLGEN
      5.  On-line Documentation (Cross Reference)                  UTLXRF
      6.  Communications/Network                                   UTLCMN
      7.  Database Utilities                                       UTLDBF
      8.  RptDist Report Distribution Facility                     UTLRPT
      9.  SplArc Spool file archive utility                        UTLSPL

  Support functions
     30.  Installation Procedures                                  UTLINS

     35.  Display NUTIL installation attributes


Selection or command
===>

F3=Exit     F4=Prompt        F9=Retrieve          F13=User Support
F10=Command Entry    F14=Work with Submitted Jobs   F18=Work with output

© Copyright Navan Limited, 2023
```

NUTIL Programmer environment

The NUTIL Programmer Environment is a set of utilities designed to assist programmer development. It consists of the following:

- An initial sign on to the environment
- 'Intelligent' compiler facilities
- The capability of interrupting a currently running process to perform another function
- A 'popup' programmer function window that can be accessed via the attention key

Installation is very simple - all you have to do is apply a minor change to your user profile.

WRKDBF Work with database files

WRKDBF is a PDM styled database file manipulation tool that allows easy access to the main operations normally performed on database files. In addition it provides file display and edit capabilities not provided by standard OS/400.

WRKDBF allows you to

- Edit and display database file data
- Copy files (either via CRTDUPOBJ or CPYF)
- Save and restore files
- Clear, Reorganize, Delete and Move files
- Start and end journalling
- Display file, field and database relationship information
- Work with Query or Data File Utilities
- Cleanse files by selection string
- Send/Receive network files

…all from the one display panel.

The EDTDBF (Edit database file) function of WRKDBF is a full screen database file update utility designed to run on the IBM iSeries. It is capable of displaying database records in both physical and logical files in either keyed or arrival sequence without any programming effort.

For performance reasons, EDTDBF supports a maximum record length of 4096 bytes.

EDTDBF allows you to

- display formatted journal data extracted by the DSPJRN command
- display records in keyed sequence
- display records in arrival sequence
- display records in hexadecimal format
- display record contents by field name
- add records by field name
- update records by field name
- copy records by field name
- delete records
- print records by field name
- window by field name
- scan records for a character string
- scan records for a hexadecimal string

EDTDBF does not support the following:

updating of Join files
updating of logical files based on multiple members

*NULL values in null-capable fields
Floating point fields
Encoded Vector Index (*EVI) files

The DSPDBF (Display database file) function of WRKDBF provides all the facilities of the EDTDBF except for the file updating capability.

GENPGM Program generator

The NUTIL Program Generator is a productivity tool for programmers at all levels within your organization. It rapidly produces proven ILE RPG (or, optionally, OPM RPG/400) code from a comprehensive set of template programs. You can extend these templates as you wish or tailor them to your standards, enabling simple seamless integration with your current systems.

The system is driven from a single menu and can be used productively by your staff within 30 minutes of installation.

The Program Generator performs best with data base files having well defined field attributes. For fields which require automatic date formatting and validation then EDTCDE(Y) or a user-defined *8 digit date* edit code must be specified to force the appropriate source code to be generated. All numeric dates are assumed to be stored on file in Year/Month/Day format, and displayed to the user in *JOB format.

Null-capable fields and floating point fields are not supported.

Date, Time and Timestamp fields are supported. In the case of date data type, dates in the range 1940-2039 are fully supported. You will need to (manually) modify the generated source code to allow the processing of dates outside this range.

General

- Simple to use
- Fully commented programs
- Consistent Code
- Creates Display/Printer/RPG/CL source
- Enforces Programming Standards
- User can extend or tailor Standard Templates

Technical

- Supports Physical/Logical and Join files
- Subfile Page at a time, using PageDown/PageUp
- Subfile positioning by key
- Interface to merge User Functions
- Automatic Date formatting & Validation
- Error Message Handling
- Record Locking Support
- Help Key Support
- Display files use Putovr Ovrdta Ovratr

Performance

For an experienced programmer to design, code and test a File Maintenance program over a file with 40 fields could take 4-5 days if the program was to include the following specifications:

- Program controlled Pagedown/Pageup
- Ability to load subfile using partial key
- Record Locking
- Trap and display/print abnormal errors

The Program Generator function within NUTIL can generate the required source code for the above function in a matter of minutes.

JOBLOGS Job log management

Normal OS/400 operation of job log processing requires retaining hundreds of logs on an output queue. Apart from resulting in a slow response when displaying the output queue, this method also increases the number of jobs in the system, and eventually affects system performance.

The aim of this system is to allow the retention of joblogs for specified number of days without holding the logs on an output queue. The logs are loaded into a database file and cross-referenced for future access. The job log is then removed from the output queue, effectively removing the job from the system.

The system operates by maintaining two joblog files: an index of the logs, and a data file which contains the text information of each job log.

The files are loaded upon request, via a command supplied. It is usually more convenient if this job is run as part of a routine, perhaps during a housekeeping routine, or as part of a regular start-up routine.

Once the job log information is in the JOBLOGS files, it can be accessed via the supplied NWRKJOBLOG command.

Routines are also provided to define how long the stored joblogs are to be retained in the JOBLOGS database. If a retention period is defined the logs will be automatically removed when they have exceeded the retention period.

Job Scheduler

SCHEDULER is an integrated system designed to run on the IBM iSeries, executing requested jobs in a batch subsystem based on a predefined job schedule. With little or no user intervention, SCHEDULER will run your iSeries all day, every day. Accordingly you achieve:

- Reductions in overtime and manpower needs
- Improved response time

SCHEDULER allows you to:

- Pre-schedule a job to run any time on any day
- allow predefined jobs to run on specific dates
- allow predefined jobs to run on a specific day every week
- allow predefined jobs to run every day, week, month or year
- schedule jobs that are dependent on other jobs
- interface easily from user programs

All jobs that are scheduled to run are listed at the beginning of the day. This may be updated and reprinted at any time during the day. An audit trail may then be produced the next day that provides you with a list of what *did* happen. Any jobs that were not run (due to system powerdown or operator intervention) are also listed.

SCHEDULER runs in its own subsystem (not QBATCH), so that normal day-to-day operations are not affected by SCHEDULER jobs tying up your production subsystems.

SCHEDULER runs jobs that have been described in a job scheduler file. Each job is defined with time and schedule frequency. At predetermined time intervals, the job scheduler file is read to see if there are any jobs to be run at this time, on this day. If there are, the job is submitted for processing in the SCHEDULER subsystem.

Jobs may be defined in one of two ways:

- a ‘once off’ job, that is to be run only once
- a standard job, that is to run on a regular, predefined frequency (every day, week,, year, every Monday, etc)

A system data area is used to define and control what days of the week are available to the Scheduler. If a day is designated as ‘not available’, no automatic scheduling will be performed on that day.

A calendar file is used to define and control what calendar dates are not available to the Scheduler. Any dates so designated will still be processed, but automatic scheduling will not be performed (manual override is still allowed if the Scheduler job is running).

Using a command supplied with SCHEDULER, loading of non-standard jobs is very similar to the standard SBMJOB (submit job) command, with additional parameters defining the scheduling requirement.

At any time of the day, the current job schedule may be viewed and maintained. Jobs may be added to the schedule, removed from the schedule, or rescheduled to a different start time. The maintenance display shows all jobs that are currently scheduled, allowing easy schedule maintenance.

Also, at any time, standard jobs can be added, maintained or deleted, by an easy to use maintenance display. This allows you to change the scheduling attributes, or to change the actual job itself. Any change to the day or time will affect the current day’s schedule of jobs immediately.

A job may include a ‘warning’ message to the System Operator, advising that the job is about to be processed. A ‘hold’ may then be placed on the job if it is not required to be run at that stage.

XREF Cross Reference utility

XREF (NUTIL Cross Reference Utility) is a programmer productivity tool providing immediate access to documentation and cross reference information. Information provided allows you to determine where (and how) a file is used, where a field is used, program summary information listing all objects accessed, and so on.

Information is presented via the screen with the option to produce a print of the displayed data.

Changes to the "LIVE" system can be reflected immediately in the cross-reference inquiries, using the tools provided in NUTIL.

XREF can be split into three broad functions:

1. Cross Reference Generation
2. Cross Reference Inquiry
3. Programmer Development Facility

Cross reference Generation and Inquiry

This function has been developed to assist the data processing department keep all systems documentation accurate and up to date. It achieves this by storing all relevant information in its own Data Base, that is created once only for a new system, and kept up to date each time an object is modified or a new object added to the production library.

Once the Cross Reference Data Base is created for an application, all objects within the application will exist in the Data Base and are available for inquiry. Using program generation facilities provided with NUTIL, the database is kept updated when application objects are altered.

The Database created in XREF is stored by an 'application' (or system) code that is specified during the generation of each application. The database will be generated for all objects that exist in the specified Object Library, and thereafter will be stored by the Object Library name for future access.

Programmer Development Facility

The Programmer Development Facility function of XREF provides a more 'intelligent' version of the Programmers menu supplied by IBM, in that it ensures that a programmer never manipulates source members in a 'Production' source file.

Also, whenever any Production source member is replaced by a newer version, a copy is 'archived' so that a previous version is always available should it ever be necessary to revert to using an older version.

The Programmer Development Facility is set up to work around an installation which has up to five types of program library:

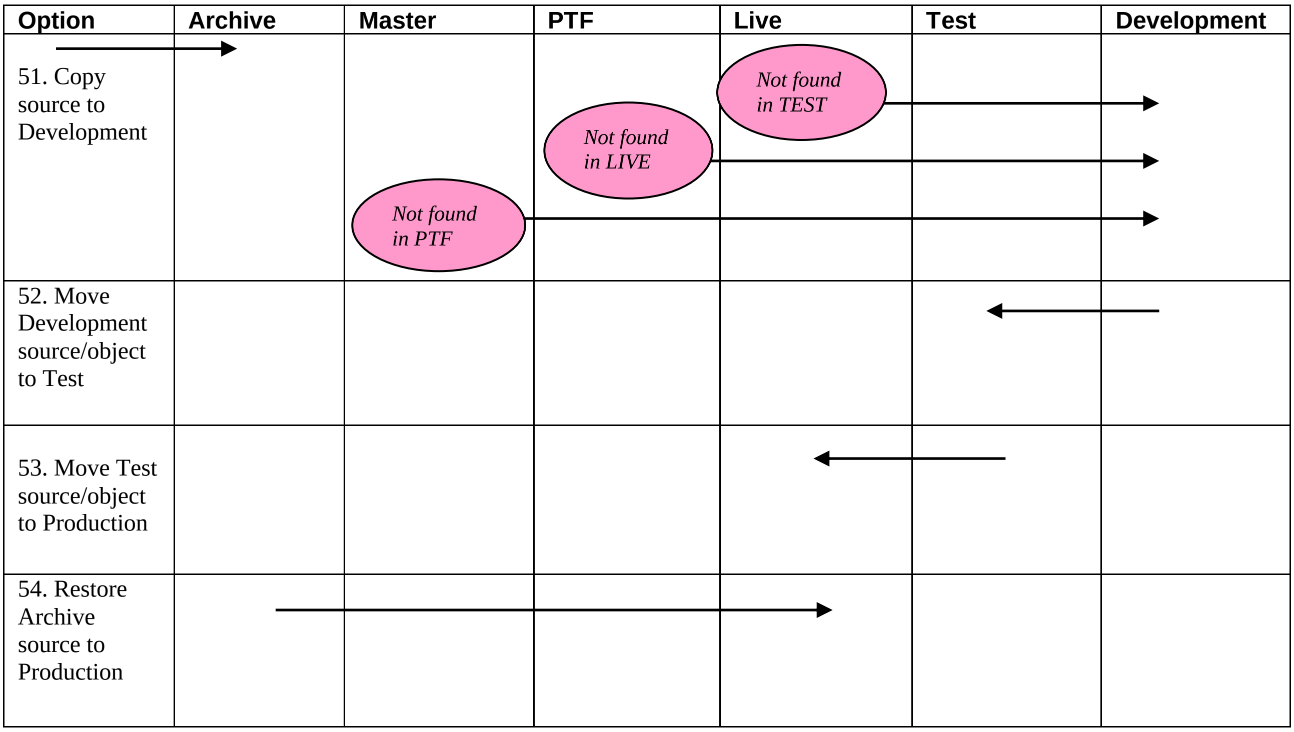
- ***Master***: These libraries contain "MASTER" versions of source code. These masters are considered as 'read-only' versions, and are never updated by this system. Master libraries are optional; you may not have this requirement.
- ***Production***: These libraries contain the "LIVE" objects. For instance you may have one library containing all the programs and one for the files, or you may have a library for each application. These are the libraries that are in use by users of the system.
- ***Test***: These libraries are used to store modifications where development has finished and testing is in progress. These are the libraries used by system verification personnel, where the modification is to be approved before being installed in the production system.
- ***Development***: These libraries are used to store modifications in progress, as well as any new development. These are the libraries used by programmers for testing and development purposes.
- ***Archive***: These libraries are used to store the prior versions of production source code. Archive libraries are optional; you may not have this requirement.

Programmers should only be allowed to manipulate source members in the Development libraries (this is the prime objective of this facility). The only time the Production source is accessed is when a new version of a program is being installed in the user’s environment. When a production source member is replaced by a new version, the *old* version is *always* archived, rather than just replaced.

Source/object flow within the development facility

It is important to understand where the Development menu option causes the source to be *moved*, rather than just copied.

The following diagram shows the movement of source and objects between the defined environments, when using the Development menu options:



Option	Archive	Master	PTF	Live	Test	Development
51. Copy source to Development		*Not found in PTF*	*Not found in LIVE*	*Not found in TEST*		
52. Move Development source/object to Test						
53. Move Test source/object to Production						
54. Restore Archive source to Production						

Option 51 shows four different source manipulations, but only one is actually performed.

> If the source is found in the Test environment, it will be *moved* back to Development (along with its associated object).

If the source member could not be found in the Test environment then a check will be made first in the Production, then the PTF and finally in the Master environments.

> If found in one of these environments, then the first one found will be *copied* from there into the Development environment.

RptDist Report Distribution Facility

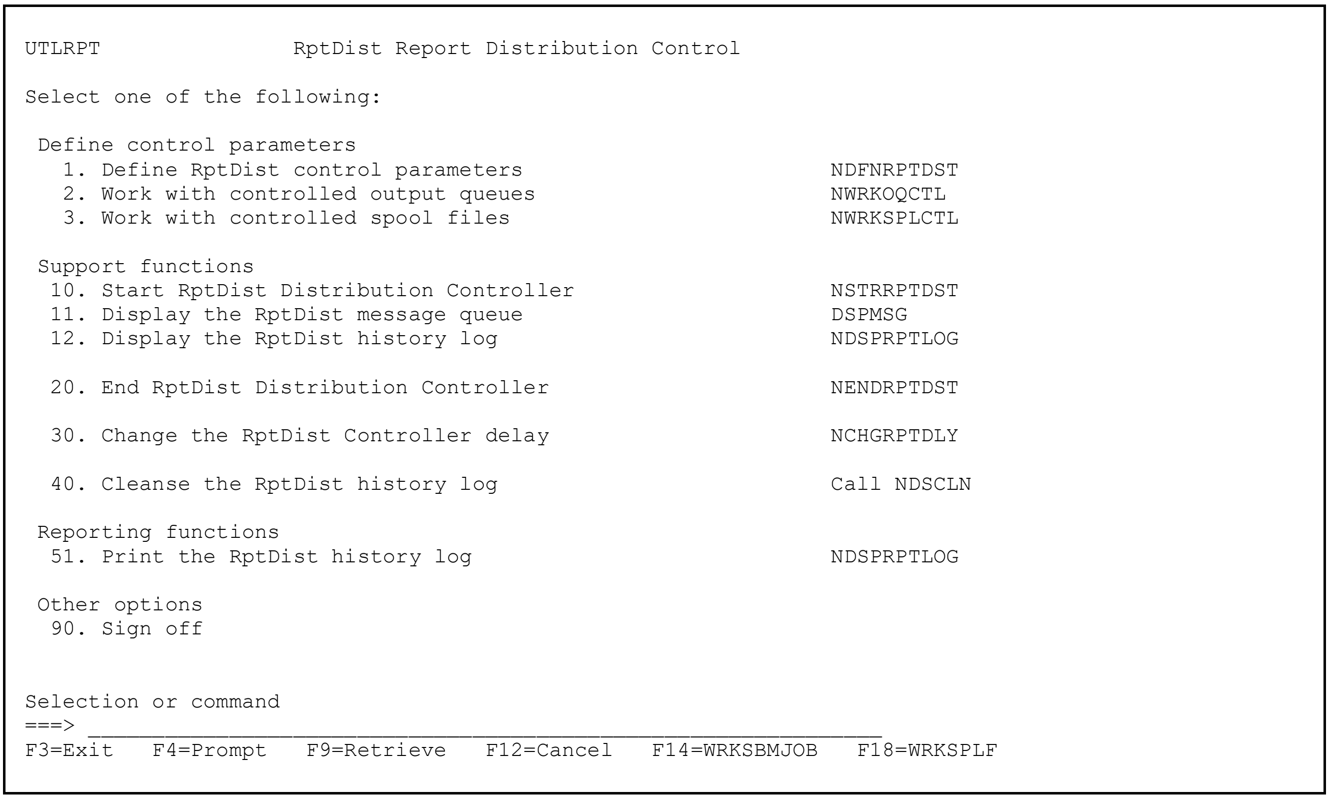
The Navan RptDist Report Distribution facility is designed to allow you to distribute a spool file to multiple users, or locations, without any special programming required on your existing user applications.

Spool files can be distributed to users on the local machine, users on other iSeries's connected to your network (via SNADS), or to users on machines other than iSeries's (via TCP/IP). The original spool file can be retained for archiving or it can be automatically removed after it has been successfully distributed.

Control is confined to specific output queues that are defined within a control list.

Specific spool file names are then defined which, combined with the list of controlled output queues, determines the spool files that will be processed by the facility.

A control interrogator job is started by the user to process the required spool file distributions. The job can be started and stopped as required, but would normally be active all day. As spool files arrive on any controlled output queue this job will scan the distribution control files and determine how to distribute the report.



```
UTLRPT               RptDist Report Distribution Control

Select one of the following:

 Define control parameters
   1. Define RptDist control parameters                        NDFNRPTDST
   2. Work with controlled output queues                       NWRKOQCTL
   3. Work with controlled spool files                         NWRKSPLCTL

 Support functions
  10. Start RptDist Distribution Controller                    NSTRRPTDST
  11. Display the RptDist message queue                        DSPMSG
  12. Display the RptDist history log                          NDSPRPTLOG

  20. End RptDist Distribution Controller                      NENDRPTDST

  30. Change the RptDist Controller delay                      NCHGRPTDLY

  40. Cleanse the RptDist history log                          Call NDSCLN

 Reporting functions
  51. Print the RptDist history log                            NDSPRPTLOG

 Other options
  90. Sign off


Selection or command
===> ______________________________________________________________
F3=Exit   F4=Prompt   F9=Retrieve   F12=Cancel   F14=WRKSBMJOB   F18=WRKSPLF
```

All RptDist functions can be performed from the RptDist main menu, shown above. It is also possible to run the commands individually, from a normal command line. Command names are shown at the right of the menu option.

Work with controlled output queues

The NWRKOQCTL command is used to define which output queues on your system that you want the RptDist interrogator job to control. Any spool files arriving on an output queue that is defined here will be eligible for automatic report distribution.

Work with controlled spool files

The NWRKSPLCTL command is used to define which spool files on your controlled output queues that you want the RptDist interrogator job to control. Any spool files defined here arriving on a controlled output queue will be eligible for automatic report distribution.

Reports can (optionally) be selectively filtered by the use of data 'masks' - if a defined area within the actual spoolfile data contains the specified match criteria, the report is distributed.

Distribution of spool files can be controlled in 4 ways:

1. You can send spool files via SNADS (SNA Distribution Services) to any user on your network
2. You can send spool files via TCP/IP to any printer queue on a remote system known to your iSeries
3. You can send spool files to another output queue on your local iSeries
4. You can send spool files (as email .pdf attachments) to internet email addresses.

You can use any, or all, of these options to control your distributions. It is possible to have one original spool file that is distributed to many network users, many remote systems and many output queues.

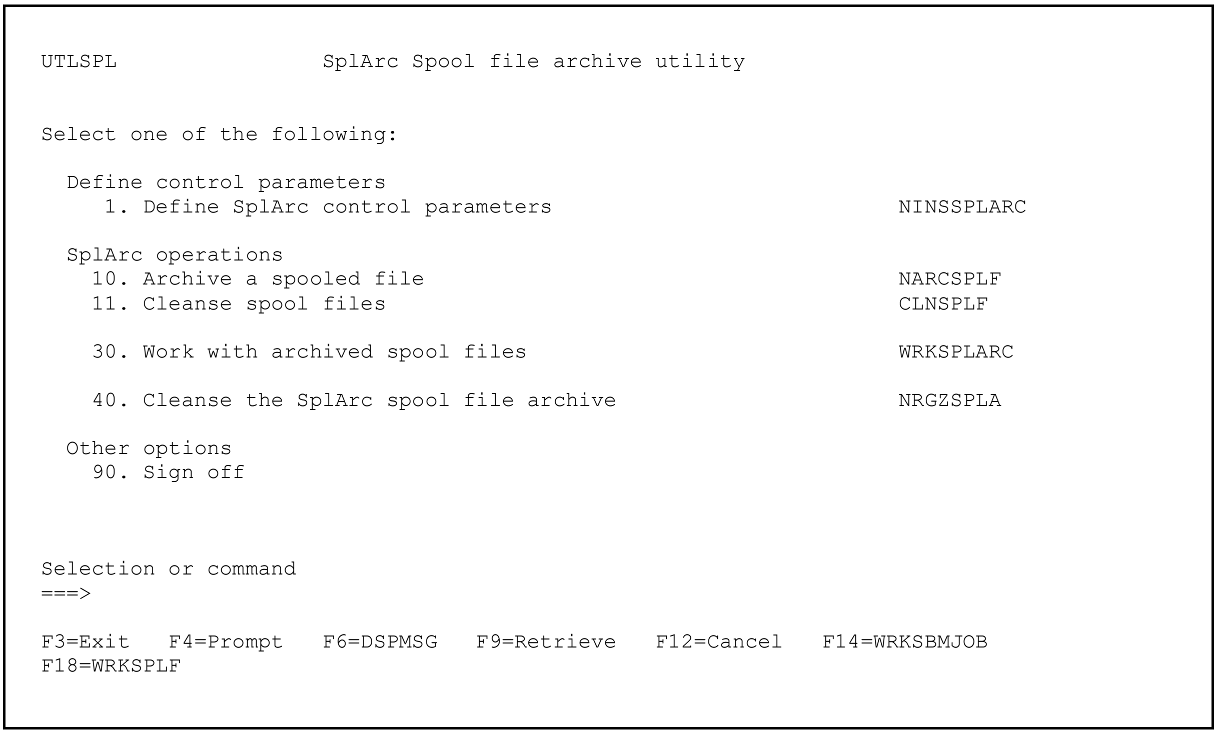
You do, however, need to define the distributions that you require. The RptDist interrogator job will only process spool file names that have been defined in the spool file control list. You do this via the NWRKSPLCTL command and define in the TYPE parameter which subsection of the spool file table you will work with:

> *NETFUSER specifies that you will be defining distributions to be made via SNADS to network users
>
> *TCPUSER specifies that you will be defining distributions to be made via TCP/IP to printer queues on remote systems
>
> *OUTQ specifies that you will be defining distributions to be made to other output queues on your local system
>
> *EMAIL specifies that you will be defining distributions to be sent via email.

SplArc Spool file archive utility

The Navan SplArc spool file archiving utility allows you to move spooled files in to archive storage. From there you can later display and generate copies of the original spooled file output.

The SplArc functions can be accessed from the GO UTLSPL menu:

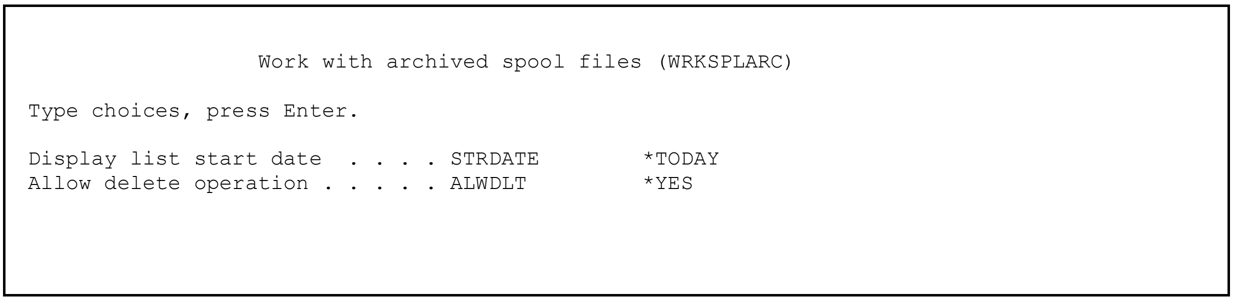


```
UTLSPL                SplArc Spool file archive utility


Select one of the following:

  Define control parameters
     1. Define SplArc control parameters                           NINSSPLARC

  SplArc operations
    10. Archive a spooled file                                     NARCSPLF
    11. Cleanse spool files                                        CLNSPLF

    30. Work with archived spool files                             WRKSPLARC

    40. Cleanse the SplArc spool file archive                      NRGZSPLA

  Other options
    90. Sign off



Selection or command
===>

F3=Exit   F4=Prompt   F6=DSPMSG   F9=Retrieve   F12=Cancel   F14=WRKSBMJOB
F18=WRKSPLF
```

Before Splarc can be used it must be installed; this can be achieved by taking option 1 from the menu.

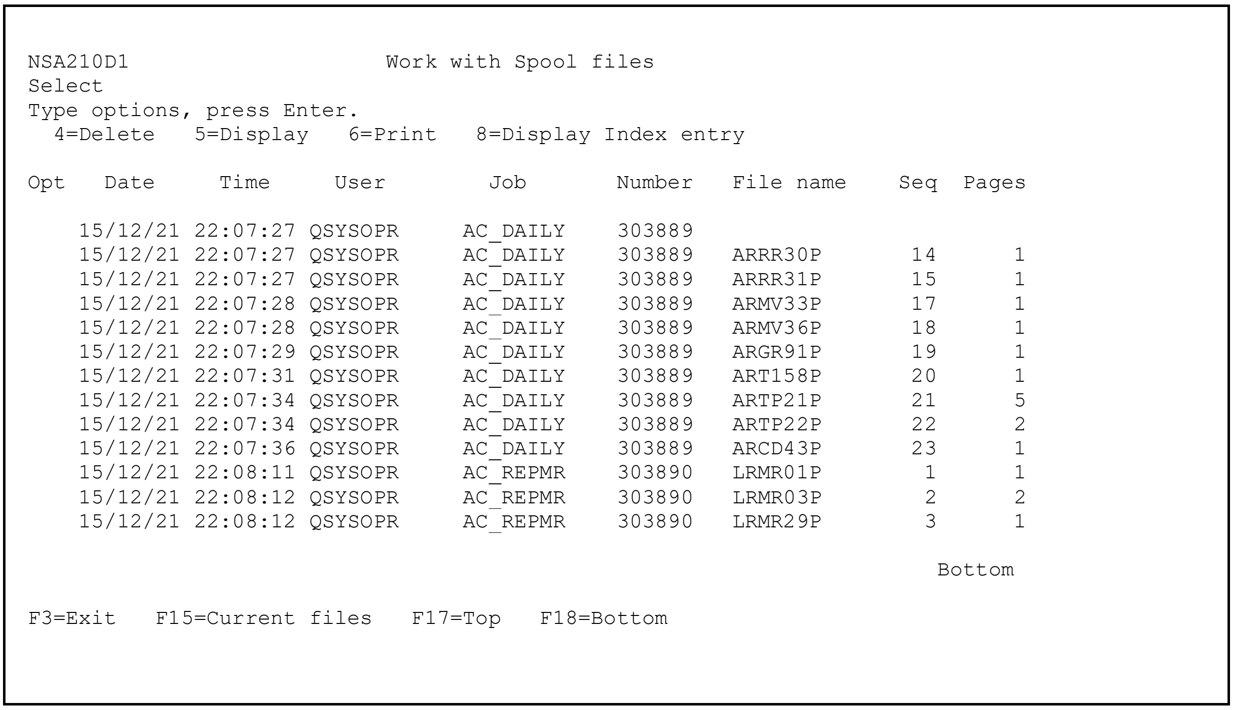
Working with archived spool files

Once spooled files are moved to the SplArc archive they are no longer considered spooled data by the iSeries system. To process them you use the WRKSPLARC (work with archived spool files) command:



```
                  Work with archived spool files (WRKSPLARC)

Type choices, press Enter.

Display list start date  . . . . STRDATE        *TODAY
Allow delete operation . . . . . ALWDLT         *YES
```

From here you will then be shown the main 'work with' panel. The list shows the spooled files currently stored in the SplArc archive. Each line in the list represents one archived spooled file.



```
NSA210D1                    Work with Spool files
Select
Type options, press Enter.
  4=Delete   5=Display   6=Print   8=Display Index entry

Opt   Date     Time     User        Job       Number   File name    Seq  Pages

    15/12/21 22:07:27 QSYSOPR    AC_DAILY    303889
    15/12/21 22:07:27 QSYSOPR    AC_DAILY    303889   ARRR30P       14      1
    15/12/21 22:07:27 QSYSOPR    AC_DAILY    303889   ARRR31P       15      1
    15/12/21 22:07:28 QSYSOPR    AC_DAILY    303889   ARMV33P       17      1
    15/12/21 22:07:28 QSYSOPR    AC_DAILY    303889   ARMV36P       18      1
    15/12/21 22:07:29 QSYSOPR    AC_DAILY    303889   ARGR91P       19      1
    15/12/21 22:07:31 QSYSOPR    AC_DAILY    303889   ART158P       20      1
    15/12/21 22:07:34 QSYSOPR    AC_DAILY    303889   ARTP21P       21      5
    15/12/21 22:07:34 QSYSOPR    AC_DAILY    303889   ARTP22P       22      2
    15/12/21 22:07:36 QSYSOPR    AC_DAILY    303889   ARCD43P       23      1
    15/12/21 22:08:11 QSYSOPR    AC_REPMR    303890   LRMR01P        1      1
    15/12/21 22:08:12 QSYSOPR    AC_REPMR    303890   LRMR03P        2      2
    15/12/21 22:08:12 QSYSOPR    AC_REPMR    303890   LRMR29P        3      1

                                                                       Bottom

F3=Exit   F15=Current files   F17=Top   F18=Bottom
```

Utility commands and modules

Also included is a set of utility commands and programs (contained in library NUTIL) that have been developed to fill in areas not covered by standard OS/400. The commands function in the same way as standard OS/400 commands, and all support the command prompt facility from a command entry screen.

The following is a listing of the main commands and modules supplied.

Utility Commands

BLDACCPTH *- Build/Rebuild Access Paths*

This command will attempt to build/rebuild an access path over the file(s) requested. This is particularly advantageous where you use MAINT(*DLY) a lot, or where the system has terminated abnormally, possibly requiring access paths to be rebuilt.

CHGJOBDLIB *- Set JOBD to current user library list*

This command allows the library list of the nominated JOBD to be the same as the current interactive library list. This is a permanent change.

CHGLIBOWN *- Change Library Ownership*

This command changes the ownership of all objects in a specified library (including the library object itself) to a new owner. Also, you can reset the public authority to the objects.

CHGSRQAUT *- Change user authority to System Request functions*

This command allows you to enable/disable system request functions for a user job. It is extremely useful in user application programs, to be able to stop a user from ending the currently running function by the method of using option 2 (end previous request) of the system request menu.

CHKACTJOB *- Check whether job is active in specified subsystem*

This command will check a specified subsystem to see whether the specified job name is currently active in that subsystem. If it is, an escape message is sent back to the caller.

CHKSRCOBJ *- Check source/object compatibility*

This audit routine will check objects in a specified object library against the source in specified source files. Reports are printed showing any incompatibilities detected.

CLNPFM *- Cleanse Physical File member*

This command will remove records from a file, based on extraction criteria provided. The extraction selection criteria is in the form of an OPNQRYF (Open Query File) QRYSLT (Query Select) statement and is bound by the same rules as that parameter. You can optionally archive the removed records to a separate archive file, allowing you to keep historical data.

***CLNSPLF** - Cleanse Spool Files*

This command allows you to perform group processing of spool files on your iSeries. Spool files can be removed or archived based on input parameters entered.

***CLRFILES** - Clear Library Physical Files*

This command allows you to clear (remove all data, including deleted records) a group of physical files in a specified library. The request may be generic, so that a request of 'GL*' will clear all gl... files within the specified library. ALL members are cleared within each file, including source files (if selected).

***CPRSRCMBR** - Compress/Decompress Source File Member*

This command allows you to compress or decompress a source file member. Compression can typically save 80% of disk storage currently held by your source files. The resulting compression ratio is reported upon successful compression of a source member.

***CPYPF** - Copy Physical Files*

This command allows you to generically copy Physical files from one library to another.

***CVTDECERR** - Report/Convert Decimal Data Errors in a file*

This command allows you to test specific, or all, fields in a physical file in order to determine whether decimal data errors exist in fields defined as numeric on the file DDS.

> An OPTION parameter allows you to either report decimal data errors encountered, update errors (by setting the field(s) in error to zeros) or both.
> A FIELDS parameter list allows you to test specific fields in the file. If you do not specify any, *ALL is the default.

***DSPFILE** - Print a Data Description Record Layout for a file*

This command allows you to print a formatted file field listing for a specified file or group of files. Options are available to print file key details, select/omit logic and also to filter in/out display files and logical files from the group.

***DSPLIBSIZ** - Display Library Size*

This command will display the total size of all objects in a specified library.

***DSPOBJINF** - Display Object Information*

This command displays an informational panel which provides you with the basic attributes of an object, in a similar way to DSPOBJD.

***DSPRCDFMT** - Display File Record Format*

This command displays file field information based on the file requested.

***DUPMSGD** - Duplicate Message Description*

This command allows you to duplicate a message file message, with the option to then modify the duplicated message.

***DUPSPLF** - Duplicate Spoolfile*

This command allows you to duplicate a spool file. The spool file named is copied and placed on a named output queue.

***EDTDTAARA** - Edit Data Area Contents*

This command allows you to edit the contents of a Data Area.

***EDTPSTDFT** - Edit PassThru Defaults*

The EDTPSTDFT command allows you to maintain the default data stored for remote machines that is used by the NPASTHR command.

***ENDMBRLCK** - End jobs holding a member lock on a specified file*

This command will start a controlled ending of any job that currently has a member lock on a specified file.

***MRGLIB** - Merge two libraries*

This command allows you to merge the contents of one library into another. The object in the 'From' library will replace a duplicate object in the 'To' library. However a 'work library' may be defined in which case duplicate objects will be moved there, rather than replaced.

***NADJSYSTIM** - Adjust system time*

The NADJSYSTIM command allows you to adjust the time on the system clock by a specified number of seconds.

***NCHGCCSID** - Change physical file CCSID*

The NCHGCCSID command allows you to alter the CCSID of a specified file, or group of files.

***NCPYDBFIFS* -** Copy Database File to IFS

The NCPYDBFIFS command simplifies the process of copying a database file member to an IFS stream file.

***NCPYFDDM** - Copy a file using DDM*

The NCPYFDDM command allows you to copy a file from this machine (the source machine) to a remote machine (the target machine).

***NCVTDBFCSV** - Convert Data file to .csv File*

The NCVTDBFPCD command allows you to convert a DB2/400 database file into a Comma Separated Variable (.csv) file.

> Optional extraction selection criteria in the form of an Open Query File query select (QRYSLT) statement can be used and is subject to the same syntax rules as that parameter. Refer to the CL Reference manual for a full discussion of the OPNQRYF (Open Query File) command.

***NCVTDBFXML** - Convert Data file to XML File*

The NCVTDBFPCD command allows you to convert a DB2/400 database file into an XML Data File file.

> Optional extraction selection criteria in the form of an Open Query File query select (QRYSLT) statement can be used and is subject to the same syntax rules as that parameter. Refer to the CL Reference manual for a full discussion of the OPNQRYF (Open Query File) command.

***NCVTQRYDFN** - Convert QUERY/400 *QRYDFN to Query Management/400*

The NCVTQRYDFN Convert Query Definition object to Query Management command allows you to convert a Query Definition, which was created using the Query/400 Program Product, into equivalent Query Management/400 source and objects. Using the *QRYDFN object specified, the command will retrieve the QMQRY and QMFORM source into the specified source files and then attempt to generate *QMQRY and *QMFORM objects based on the retrieved source.

***NCVTRPGSRC** - Convert RPG/400 to ILE RPG source code*

The NCVTRPGSRC command allows you to convert RPG III or RPG/400 source code to ILE RPG source code. In addition to the processing done by the IBM CVTRPGSRC command, this command converts field names from upper to lower case and also generates Data specifications ('D' specs) to define any fields that were previously defined in calculation specification lines.

***NCVTSPLSTM** - Convert spooled file to stream file*

The NCVTSPLSTM command lets you convert a spooled file in to a stream file, to be stored in the iSeries integrated file system (IFS) root file system.

Using shared folders, the resulting stream file can be opened in a PC application such as a document viewer, spreadsheet or a browser. The stream file can also be sent as an email attachment, published on the web, etc.

***NDLTDBR** - Delete database file relations*

The NDLTDBR command will delete any files that are dependent upon the file specified. A report will be printed showing the dependencies that existed before the deletion processing commences.
Note: The specified file itself is not deleted, only the files that are dependent upon it.

***NDLTOBJ** - Delete objects by object type*

This command allows you to delete a group of objects by a specific object type. For example, deletion of all RPG programs within a specific application. Also deletions may be generic, such as gl* will delete all objects of the specified object type within a specified library.

***NDUPOBJ** - Duplicate objects*

This command allows you to duplicate objects. It is similar to the CRTDUPOBJ command, but has the added advantage of optionally deleting any existing object in the destination library.

***NPASTHR** - Start Remote Connection*

The NPASTHR command is a 'front-end' command to the standard iSeries remote connection commands (STRPASTHR and STRTCPTELN), in that it obtains the communications parameters for the relevant command from a user-defined file.

NPASTHR starts each connection (Pass-through only) session as a 'group job' to allow you to have multiple sessions open to multiple machines at the same time. Another advantage of using the group jobs concept is that you can suspend the pass-through sessions on the target systems.

> If you use the TFRPASTHR command to return from the target system back to the source system, the pass-through session remains open (suspended mode). When you next enter the NPASTHR command again for that same System ID you will be returned to the suspended target session (provided the target job is still available).

***NPRTDBFINF** - Print database information*

The Print database file information (NPRTDBFINF) command provides a report detailing selected information retrieved from the attributes of a specified physical database file. Details of all dependent files are also printed on the report.

***NPRTJRN** - Print Journal Images*

The Print Journal (NPRTJRN) command provides a 'front-end' to iSeries system journalling, in that it will provide formatted listings of journal entries processed within a specified timeframe. It can work in one of two ways:

> By specifying a command to process, it will run the command and then print listings of all journal entries caused by that processing
>
> If no command request is specified (the OS/400 Display Journal command is prompted), you can print prior journal information from other requests that have been processed

The record images are formatted and printed using the file field definition database that is used by the NUTIL Work with Database files (WRKDBF) utility.

***NPRTJRNHST** - Print Journal Images*

The Print Journal (NPRTJRNHST) command provides a 'front-end' to iSeries system journaling, in that it will provide formatted listings of entries processed within a specified timeframe.

The command is similar to (and uses the same processing engine as) the NUTIL NPRTJRN command. However this command can be used in all processing environments on the iSeries, whereas the NPRTJRN command can only be used interactively.

***NRPLSTR** - Scan and Replace character string in Source file*

This command works in a similar way to the NSCNSTR command previously described, but in this case, where a match is found on the SCAN string it is replaced using the contents of the REPLACE string.

***NRTVJOBDA** - Retrieve *JOBD Attributes*

The Retrieve Job Description Attributes (NRTVJOBDA) command is used in a CL program to retrieve the value of one or more attributes defined within the job description and place the values into the specified CL variable.

***NRUNSQL** - Run SQL Statement*

The Run SQL Statement (NRUNSQL) command allows you to run an Interactive SQL statement, without having the iSeries SQL/400 Program Product installed.

NSCNSTR *- Scan source for a character string*

This command allows you to scan a source member for a defined character string. Multiple searches are allowed and multiple members may be searched. A new enhancement allows the use of the OPNQRYF facility, by defining the search string as the QRYSLT (Query select) parameter.

A report lists all occurrences found of the specified scan string(s).

NUPDRPGSRC *- Update (Enhance) ILE RPG Source Code*

The NUPDRPGSRC command allows you to 'reprocess' your ILE RPG source code to potentially improve its appearance and readability.

RGZFILES *- Reorganise Library Physical Files*

This command allows you to reorganise multiple files in a library with one command. ALL members in a selected physical file will be eligible for reorganisation. Reorganisation will only occur if the percentage of deleted records equals or exceeds the requested parameter.

RGZSRCMBR *- Reorganise Source Member*

This command allows you to reorganise the information in a source file member. Line numbering can be re-sequenced, source dates reset.

RPGINDLST *- Print Indented RPG source listing*

This command will print out a formatted RPG source listing of a complete, or specified portion of a source member. As well as having the indent level number printed (in the same way as a standard compile listing), all *IF, CAS, SELEC* and *DO* groups will be physically indented for ease of reading.

SNDNETFMBR *- Send Network File Member*

This command allows the sending of one, or many, members in a file via SNADS. The SNDNETF command is performed for each member within the generic group (and source attribute, if specified) requested. Selection of the member to send is also available.

TIME *- Display system time*

This command allows you display the current system time and date.

WRKSAVF *- Work with Save File*

The Work with Save File (WRKSAVF) command allows you to view the contents of a Save File and selectively restore objects from it.

Utility Programs and modules

In addition to the commands supplied, callable subprogram modules are also included in NUTIL. These include:

@CHKDAT	Check input date for validity
@CHKDAT8	Check 8 digit input date for validity
@CHKTIM	Check input time for validity
@CLCLOG	Calculate logarithm
@CLCDAT	Calculate date, given a start date and duration to add/subtract
@CVTCDT	Convert *CYMD from/to *ISO date
@CVTJUL	Convert a *YMD date from/to Julian format
@ERRMSG	Error message handling program
@GETDAT	Calculate a date, given a date and number of days
@GETDAYS	Get the number of days between two dates
@GETWKNR	Get the *ISO week number of the year for a given date
@RANDOM	Random Number Generator
@RTVASI	Retrieve auxiliary storage information
@RTVDAT	Retrieve system date/time
@RTVDAT8	Retrieve system date/time (*ISO format)
@RTVJDT8	Retrieve current 8 digit date/time (*JOB format)
@RVSDAT	Convert a date from one format to another
@RVSDAT8	Convert an 8 digit date from one format to another
@RTVDTS	Retrieve *ISO timestamp
@RTVIPA	Retrieve IP address for TCP/IP workstation
@SNDSTS	Status message handling program

And many more (refer to NUTIL User Guide for the full listing).

The modules are also available via a single service program (NSRVPGM) for ILE processing. Binding Directory NBNDDIR is also available for linking the service program to your applications.

Contacts

Europe, Middle East and Africa

Navan (UK) Limited
The Courtyard
Brooklands   Broughton
Milton Keynes   MK16 0HU
England

Email: helpdesk@ navan.co.uk

Rest of the World

Navan Australia Pty Ltd
22 Torrington Road
Maroubra
NSW   2035
Australia

Email: helpdesk@navan.com.au

http://www.navan.co.uk