

# RUNC Version 4.2

## Easy Commands for the ISPF Edit User

---

*By Lionel B. Dyck*

### Table of Contents

Change History .....	3
Introduction .....	4
What is RUNC? .....	4
What benefit does RUNC provide? .....	4
RUNC .....	4
RUNC Command Syntax.....	5
RUNC Command Options.....	5
RUNC Defaults.....	7
Format for Commands.....	7
Command Types and Syntax.....	8
Documentation .....	11
Symbolics .....	12
Selecting Records to Execute under RUNC.....	13
Using the RUNC command.....	13
Using the <START> and <END> statements .....	13
ERC .....	14
ERC Command Syntax .....	14
Samples .....	15
TSO .....	15
z/OS.....	16
EDIT .....	16
JES2 .....	17
JES3 .....	17
OMVS .....	17
Documentation .....	18



## Change History

Starting with Version 3.2:

Version	Date	Description
4.2	04/18/2017	Added: <ul style="list-style-type: none"><li>- EJES ST and EJES ST xxx</li><li>- SDSF ST and SDSF ST xxx</li></ul>
4.1	03/27/2017	Add: <ul style="list-style-type: none"><li>-I option to display each command results individually if a range or – ALL is requested</li><li>&lt;EXIT rc&gt; to test the return code from the current command and cease processing if the return code is equal or greater.</li><li>Add &lt;Start&gt; and &lt;End&gt; statements to define a range for processing.</li></ul>
4.0	03/21/2017	Added documentation that will appear in a popup. Using tags <DOC>, <TITLE>, and </DOC> Support Macro and Parm on both Edit and View, Including Edit Member
3.9	02/22/2017	Add new option –BS to toggle the bypass prompting for RUNC supported symbolics
3.8	02/21/2017	Add doc on use of MEMBER(x) option with Browse, Edit and View
3.7	02/17/2017	Add information on Browse, Edit and View use of VOLUME keyword
3.6	01/31/2017	Bug fixes to the code
3.4	01/26/2017	Add <ul style="list-style-type: none"><li>–L option to log messages to a dataset based on installation default</li><li>-LD to log to a sequential dataset</li><li>-LP to log to a pds member</li><li>-LX to override any log option defaulted to in the –O settings</li><li>-C to display all log datasets</li><li>%RUNCLOG used by –C option</li></ul>
3.3	01/10/2017	Add new –D(prefix) option Add &DATE symbolic (same as &UPDATE – mm/dd/yy) Change symbolic JULIAN to JDATE to conform to other date symbolics
3.2	01/03/2017	Added &RDSN, &RMBR and &RMEM pre-defined symbolics

## Introduction

### What is RUNC?

RUNC is short for Run Command. It is an ISPF Edit command that will execute many different types of commands that are stored within a dataset, or member. The dataset or member must be opened using ISPF Edit (or View). RUNC supports almost two dozen pre-defined symbolics as well as user defined symbolics to enable the commands to be generalized.

ERC, or Execute Run Command, is an ISPF command that can be used to access the dataset, or member, where the commands are stored. This is provided to simplify access without requiring the user to navigate multiple ISPF panels to get to the command dataset.

### What benefit does RUNC provide?

The benefit is that the dataset, or member, containing the commands can include comments to document the commands in the dataset, or member. The dataset, or member, could also be JCL or a document in which commands are included as comments.

For example, a document on how to allocate a partitioned dataset to contain JCL, could have an explanation of what a partitioned dataset is, how to calculate the size, and how to calculate the number of directory blocks, or whether to use a PDSE instead. Along with the text the TSO commands could be included in the dataset, or member, to perform the allocation. There could be a sample command for a simple PDS and one for a PDSE. The user can then use RUNC to execute the allocation command for the PDS or PDSE that they are interested in without having to copy and paste the command, or having to retype it.

For systems programmers and operators, a PDS could be created with members for different functions. For example, a member called ZOS could be created with many frequently, and infrequently, used z/OS commands. Using this member the systems programmer, or operator, would be able to find the command they need to use and easily execute it. The members could be more granular with one for GRS, one for SMF, one for JES2, or JES3, etc. Then using the ERC command (see below) the user would have quick access to the desired member.

## RUNC

This section will discuss how to use the RUNC command and how to code the commands in the dataset or member.

RUNC consists of an ISPF Edit command that processes records selected using C or S for an individual record or pairs of CC or SS for consecutive records. Excluded records are always ignored. A filtering option with RUNC, using the ONLY keyword, can also be used alone, or in conjunction with excluded records.

## RUNC Command Syntax

RUNC is an ISPF Edit command that is used on the ISPF Edit command line. There are several options that can be specified on the command and all of them are optional.

### RUNC Command Options

Option	Description
<b>Blank</b>	Default processing
<b>? or HELP</b>	Display the ISPF Tutorial for RUNC
<b>N</b>	The number of leading characters to skip over to find the command. Useful to skip over comments in the record. The default is 0.
<b>-A or -ALL</b>	Process ALL commands found in the data.
<b>-B</b>	ISPF Browse the messages generated by the executed commands
<b>-BS</b>	Toggle bypassing the prompt for RUNC supported symbolics
<b>-C</b>	Display a list of all LOG datasets created by RUNC
<b>-D</b>	Insert the messages generated by the executed commands in the edit data after the command as permanent records.
<b>-D(prefix)</b>	If a prefix is specified then it will have leading and trailing blanks removed and will be inserted before the messages with an intervening space. (e.g. -D(>))
<b>-I</b>	Display the command messages for each command as they are processed.
<b>-L</b>	Log all messages to a dataset.  The installation defines whether the dataset is a sequential dataset which results in one per RUNC use, or a PDS where there is one PDS per day with a member for each RUNC use.  The dataset name is hlq.RUNC.Dyymmdd.Thhmmss.LOG for sequential or hlq.RUNC.Dyymmdd.LOG for a PDS.  See -LD and -LP for more information
<b>-LD</b>	Log to a sequential dataset  One per RUNC usage
<b>-LP</b>	Log to a PDS member  One PDS per day and one member per RUNC usage
<b>-LX</b>	Turns off logging if logging was enabled by default (use of -O) for the user
<b>-N</b>	Insert the messages generated by the executed commands in the edit data after the command as notelines.
<b>-O</b>	Display panel to review or update user RUNC defaults

Option	Description
<b>-R</b>	Toggles the installation default. If the default is to Always generate a report even if the command does not generate messages then the toggle will turn that off, and vice versa. Some commands, notably ISPF and some OMVS, do not generate any messages.
<b>-S nnn</b>	Stop execution if the return code is greater than nnn. This only applies when executing multiple commands in one RUNC invocation.
<b>-T</b>	Do not capture any of the generated messages so that the messages generated by the commands display on the terminal.
<b>-V</b>	ISPF View the generated messages from the executed commands. This is the default.
<b>-W nnn</b> <b>-W W</b>	Works with <b>-D</b> and <b>-N</b> to wrap messages longer than nnn when inserting them into the data. When <b>W</b> is used then the wrap value is set to the actual Edit Data Width.  The wrap will occur at the nearest blank or at the exact character location depending on the data.
<b>ONLY(xxx)</b>	Limits the commands to be selected to only those with xxx in the command.
<b>PREFIX(xxx)</b>	Prefix all TSO commands with xxx. This only works for TSO commands.
<b>SUFFIX(xxx)</b>	Suffix all TSO commands with xxx. This only works for TSO commands.

RUNC ignores all records that are hidden using the ISPF exclude (X) process. When used with **-A**, or **-ALL**, this can be an easy way to bypass unwanted commands without having to select each individual command or command ranges.

### Notes

1. **-T** will override **-D**, **-I**, **-L**, **-N**, and **-R**
2. **-D** and **-N** will override **-B**, **-I**, **-L**, **-T**, and **-V**
3. **-L** logging dataset type is defined in the RUNC exec at installation time.

## RUNC Defaults

The first time that RUNC is used an ISPF Panel will be presented to the user to set the RUNC defaults that will be used. This Panel can also be requested using the `-O` option when using RUNC.

```
----- RUNC User Review/Update Defaults -----
Command ==>

  N (Y/N)   -B Browse the generated report of all messages
  N (Y/N)   -T No report - all messages to the terminal
  Y (Y/N)   -V View the generated report of all messages
  N (D/P/N) -L Log all messages to a permanent dataset
  N (Y/N)   -D No report - all messages inserted as Data records
             Data Prefix:
  N (Y/N)   -N No report - all messages inserted as Notelines
  Y (Y/N)   -R Generate a report of all messages
  N (Y/N)   -BS Bypass prompt for supported symbolics
  N (Y/N)   -I Display Individual Command messages
  0 (###)   -S Stop multiple command processing if return code exceeds nnn
  0 (###)   -W Wrap -D and -N messages to nnn characters (W = Data Width)
  (Y/N)     Save as permanent defaults

Notes:      1. -D and -N override -B -I -T -V and -L
            2. -T override -D -I -N -R and -L

          Press PF3 to continue
```

This figure also shows the default options used by RUNC which are to View any messages generated by the selected command(s) and to not generate a report if the selected commands do not generate any messages.

The use of `-R` is useful to generate a report that can be retained, using ISPF Create or Copy, for audit or reference purposes. Note that some commands do not generate any messages, for example ISPF dialogs, so the only message in the report would be the command start and command return code.

## Format for Commands

All commands are entered in one, or more, records. Using the following syntax:

Comments start in column 1. Supported comment formats are:

```
*
/*
/**
//
```

Use `RUNC #`, where `#` is the number of characters to skip over to bypass the comment, to execute commands that are within comments.

The command data can be numbered or un-numbered as the sequence numbers will be ignored.

## Command Types and Syntax

Below is a table documenting the supported command types and their supported syntax. The command type can be upper, lower, or mixed case.

Command Type	Syntax
<b>BROWSE</b>	<p>Browse a dataset:</p> <p>BROWSE Ddataset(dsname)            BROWSE Dataset(dsname) Volume(volser)            BROWSE Member(member-name)</p>
<b>EDIT</b>	<p>Any ISPF Edit command</p> <p>edit x all            edit x all;find &amp;string all</p> <p>Note: The command will also affect the record with the command</p> <p>Or to edit a dataset:</p> <p>EDIT DATASET(xxx)            EDIT DATASET(xxx) VOLUME(volser)            EDIT Member(xxx)            EDIT Member(xxx) MACRO(macro)            EDIT Member(xxx) MACRO(macro) PARM(parm)            EDIT Dataset(xxx(member))            EDIT Dataset(xxx(member)) MACRO(macro)            EDIT Dataset(xxx(member)) MACRO(macro) PARM(parm)</p>
<b>EJES</b>	<p>(E)JES JES2/JES3-or-z/OS-command</p> <p>ejes d ts,l            ejes d smf            ejes d smf;d asm            ejes st            ejes st xxx</p>
<b>ISPF</b>	<p>SELECT . . .</p> <p>select panel(isp@mstr)            select cmd(%xyz option1 option2)            select pgm(iefbr14) parm(abc)</p> <p>ISPExec . . .</p> <p>Ispexec libdef . . .</p>

Command Type	Syntax
<b>OMVS</b>	OMVS command  omvs ls -la omvs pwd;id omvs su;cat /etc/rc
<b>SDSF</b>	SDSF JES2/JES3-or-z/OS-command  sdsf d ts,l sdsf d smf sdsf d smf,d;asm sdsf st sdsf st xxx
<b>TSO</b>	Just enter the command on the record  lista sta st alloc ds(test.pds) new spa(15,15) tr recfm(f b) + lrecl(80) blksize(0) free . . . altlib . . .
<b>TSO Console</b>	CONS console-command  cons d ts,l cons d smf cons d asm
<b>VIEW</b>	View a dataset:  VIEW DATASET(xxx) VIEW DATASET(xxx) VOLUME(volser) VIEW Member(xxx) VIEW MEMBER(xxx) MACRO(macro) VIEW MEMBER(xxx) MACRO(macro) PARM(parm) VIEW Dataset(xxx(member)) VIEW Dataset(xxx(member)) MACRO(macro) VIEW Dataset(xxx(member)) MACRO(macro) PARM(parm)
<b>Documentation Popup</b>	<DOC> must be on a record by itself to begin the documentation section </DOC> must be on a record by itself to end the documentation section <TITLE> followed by up to 32 characters and on a record between the <DOC> and </DOC> records will be used for the popup title. Documentation records must be between the <DOC> and </DOC> records and may be up to 72 characters in length.
<b>Exit Return Code Check</b>	<EXIT rc> must be on a record by itself and the rc will be compared to the last executed command return code is equal to or greater than the rc in the exit test then RUNC processing will cease.

Command Type	Syntax
<b>Start and End Range</b>	<p>&lt;START&gt; must be on a record by itself to begin the start of the range.</p> <p>&lt;END&gt; must be on a record by itself to end the range.</p> <p>Any valid RUNC command may be between these two statements.</p>

The Edit, (E)JES, SDSF, and OMVS commands allow multiple commands on the same record separated by a semi-colon ';':

The use of the Member option with Browse, Edit, and View, will perform that action on the requested member in the active dataset. The active dataset volser will be automatically added to the action.

Commands can continue to the next record by ending the record with a '+':

```
omvs su; +
ls -la; +
pwd
```

The use of all commands is dependent upon the level of security authorization that the user has. (E)JES, SDSF, OMVS, and TSO Console commands may be restricted using the sites security package. Some TSO commands may also be restricted. The key is, if the user can use the command outside of RUNC then they can use it with RUNC.

Comments may be included on the record with the command. Comments must start with a /\* and there is no requirement for a closing \*/.

Comment example:

```
LISTD &dataset /* this command will invoke the TSO LISTD and prompt for a dataset
```

Symbolics, discussed below, are supported on all commands and in the documentation popup statements.

## Documentation

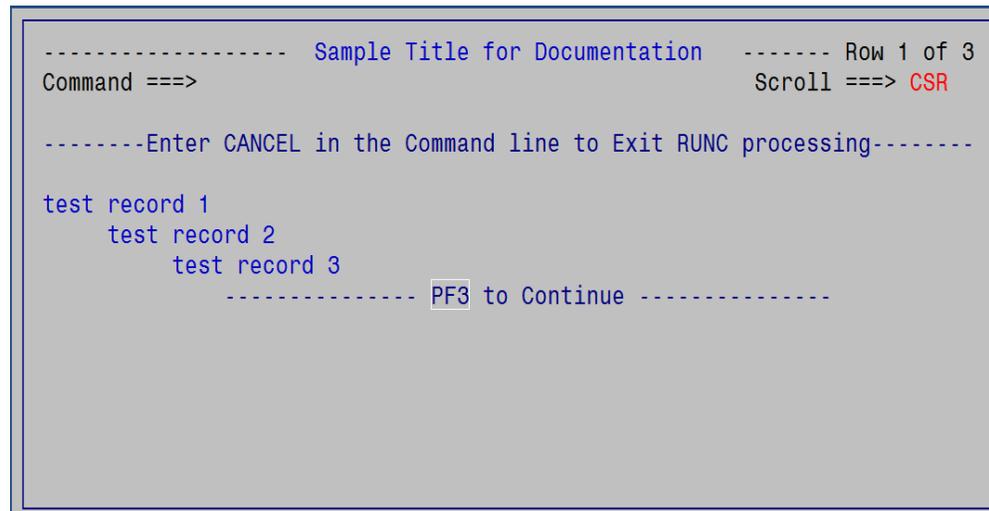
The documentation popup is a tool that can be used with RUNC –ALL to help guide the user through a series of commands or processes.

A sample member with documentation:

```
cons d iplinfo
<doc>
<title>    Sample Title for Documentation
test record 1
    test record 2
        test record 3
</doc>cons d smf
```

In the example above are two console commands with a sample documentation popup text in between.

The sample documentation popup will look like this:



The screenshot shows a documentation popup window with a grey background and a blue border. The title bar reads "Sample Title for Documentation" and "Row 1 of 3". Below the title bar, there is a "Command ==>" field and a "Scroll ==> CSR" field. The main content area contains the following text: "-----Enter CANCEL in the Command line to Exit RUNC processing-----", "test record 1", "test record 2", "test record 3", and "----- PF3 to Continue -----".

Use of PF3 closes the popup and RUNC processing continues to the next command.

To cancel further RUNC processing, enter CANCEL in the ISPF command field.

This can be very useful in a DR situation where the individual performing the DR may not be that experienced but they can follow instructions. Documentation popups can be used liberally with the required commands such as EDIT, CONS, EJES/SDSF, etc. to guide the individual in performing the DR tasks.

Symbolics may be used in both the title and documentation records. Be aware that a symbolic may extend the title (32 characters) or text (72 characters) beyond the supported limits.

## Symbolics

All commands can include one, or more, symbolics. A symbolic is a 1 to 8 character word beginning with an &. If a double & is found, e.g. &&, then the first will be retained as part of the command text.

When a symbolic is found it will cause RUNC to present to the user an ISPF Panel with the symbolic and a field to enter the value to be used for that symbolic. If a symbolic is used more than once the user is only prompted once.

RUNC supports a number of pre-defined symbolics that will be pre-filled in for the user:

Symbolic	Description
<b>&amp;DATE</b>	Date mm/dd/yy
<b>&amp;DD</b>	2 digit Day of Month
<b>&amp;DOY</b>	Day of year (e.g. 001)
<b>&amp;EDATE</b>	European Date (dd/mm/yy)
<b>&amp;HHMM</b>	Hours and Minutes
<b>&amp;JDATE</b>	Julian Date (e.g. 16001)
<b>&amp;LPAR</b>	Active LPAR Name
<b>&amp;MM</b>	2 digit Month
<b>&amp;MONTH</b>	Month (e.g. January)
<b>&amp;ODATE</b>	Ordered Date (yy/mm/dd)
<b>&amp;PREFIX</b>	Current TSO Prefix
<b>&amp;RDSN</b>	Current dataset name
<b>&amp;RMBR</b>	Current member name
<b>&amp;RMEM</b>	Current member name
<b>&amp;SDATE</b>	Standard Date (yyyymmdd)
<b>&amp;SMFID</b>	Active SMFID for the LPAR
<b>&amp;SYSNAME</b>	Active LPAR Name
<b>&amp;SYSID 1</b>	Character System Name
<b>&amp;SYSPLEX</b>	Active SYSPLEX Name
<b>&amp;SYSPREF</b>	Current TSO Prefix
<b>&amp;SYSSMFID</b>	Active SMFID for the LPAR
<b>&amp;SYSUID</b>	Current userid
<b>&amp;UPDATE</b>	USA Date (mm/dd/yy)
<b>&amp;USERID</b>	Current userid
<b>&amp;USERIDL</b>	Current userid lower case
<b>&amp;WEEKDAY</b>	Day of Week (e.g. Monday)
<b>&amp;YY</b>	2 digit Year
<b>&amp;YYYY</b>	4 digit Year

## Selecting Records to Execute under RUNC

### Using the RUNC command

In the Edit session each record that has a command is eligible to be executed. Those records that are excluded and either are not comments or for which the 'n' option was used to skip over the comments, may be executed. The way they are executed is by selecting the row by:

1. Select an individual row using the line selection option of either C or S.
  - a. C for command or S for Selection – they are synonymous
2. Select a range of rows using CC for the first and CC for the last. Or SS if you prefer.
3. Use the -A, -AL, or -ALL, option on RUNC
  - a. RUNC -A
4. Use the ONLY option on RUNC in combination with -A or -ALL
  - a. RUNC -A ONLY(XYZ)
5. Exclude all rows that you don't want to execute and then use -A or -ALL

### Using the <START> and <END> statements

Define a range of statements to be processed by using the <START> statement followed by any valid RUNC supported command and ending with the <END> statement.

Note that the <START> record must be selected using any of the prior sections methods.

## ERC

ERC was developed to speed access to the commands dataset or member without forcing the user to navigate multiple ISPF panels. ERC can be used from any ISPF Panel as it is implemented in the site ISPF command table.

The first time ERC is used the user will be prompted to enter the name of a command dataset, typically a PDS. This PDS must already be allocated. It can be Fixed Block or Variable Block and an LRECL of your choice.

If the dataset name does not begin with the users prefix then the dataset, or member, will be opened using ISPF View. This is to enable a shared command dataset and prevent accidental updates.

If the dataset name does begin with the users prefix then the dataset, or member, will be opened using ISPF Edit.

### ERC Command Syntax

Option	Description
<b>Blank</b>	Default processing
<b>? or HELP</b>	Display the ISPF Tutorial for RUNC
<b>/SET</b>	Prompt to change the command dataset
<b>-E</b>	Open using Edit
<b>Member-name</b>	A member name or member mask.  If a specific member then that member will be opened. If a mask then the member list will be presented.

## Samples

Some sample commands.

### TSO

\* list current allocations (status history sysnames)

```
lista sta hi sys
```

\* listd dataset

```
listd +          /* prompt for a dataset for listd  
      &dataset
```

\* listd dataset and members

```
listd &dataset mem
```

\* Allocate a FB PDSE

```
alloc ds(&pdsedsn) new spa(&prim,&sec) tr +  
      recfm(f b) lrecl(&lrecl) blksize(0) +  
      dsntype(library,2) maxgens(&maxgen) dsorg(po)
```

\* Backup sample.exec and sample.panels

\* \*\* requires the pdsegen application

```
del sample.exec.j&julian
```

```
%pdsegbak sample.exec sample.exec.j&julian backup
```

```
del sample.panels.j&julian
```

```
%pdsegbak sample.panels sample.panels.j&julian backup
```

## z/OS

```
* z/OS Commands Sample Command Member
* ejes is used for action commands (change to sdsf if needed)
* cons is used for display commands (change to sdsf if needed)
*
** If your shop uses ejes invoke RUNC -2 on the following command
* edit c ejes sdsf all
* edit c cons sdsf all

* Display IPL Info
cons d iplinfo
cons d u,iplvol

* Display MVS Allocation Settings
cons d alloc,options

* Display Catalog Info
ejas f catalog,report,cache
ejas f catalog,report,performance
```

## EDIT

```
//JOBNAME JOB ACCOUNT,&USERID,
// NOTIFY=&USERID,MSGCLASS=R
//*
/** TO USE THIS JCL ISSUE THE EDIT COMMAND RUNC 4 ON THE NEXT 2
RECORDS
/** EDIT C EXECLIB &EXEC ALL
/** EDIT C TSOCMD &TSOCMD ALL
/**
/** NOTE: THE EDIT COMMAND WILL ALSO BE CHANGED BY THE RUNC
/**
/**TMP EXEC PGM=IKJEFT1B,DYNAMNBR=50
/**SYSEXEC DD DISP=SHR,DSN=EXECLIB
/**SYSTSPRT DD SYSOUT=*
/**SYSTSIN DD *
TSOCMD
/*
```

## JES2

```
* Sample JES2 operator commands
*
** Prefix with either EJES or SDSF depending on which you use
*
** If your shop uses ejes invoke RUNC -2 on the following command
*   edit c sdsf ejes all
*

*   Display JES2 job definitions
sdsf $djobdef

*   Display all jobs which occupy more than 5% of the spool
sdsf $djqr,spl=(%>5)

*   Display all jobs on the print/punch queue more than 5 days old
sdsf $djqr,a>5,q=ppu
```

## JES3

```
*
** JES3 Commands
*
*
** Prefix with either EJES or SDSF depending on which you use
*
** If your shop uses ejes invoke RUNC -2 on the following command
*   edit c sdsf ejes all

*   Display Main Processor info
ejes *i,main=all
ejes *i,main=&main

*   Display Active Jobs
ejes *i,a
ejes *i,a,&main
ejes *i,j=&jobname
ejes *i,j=&jobnumber
```

## OMVS

```
* ----- *
* This is a set of sample omvs commands      *
* to demonstrate some of the capabilities.    *
* ----- *

* find a file by name
omvs su;find &dir -name '&filename'

* find file in /u/files.all
omvs grep '&string' /u/files.all
```

## Documentation

```
cons d iplinfo
```

```
<doc>
```

```
<title>Title Text on &lpar
```

```
This is a test &date on lpar &lpar
```

After viewing this documentation SYS1.PARMLIB member IEASYS00 will be presented in ISPF View.

```
</doc>
```

```
view dataset('sys1.parmlib(ieasys00)')
```