

ASG-Zeke™ Scheduling for z/OS Reference Guide

Version 6.2
Publication Number: AZM0400-62
Publication Date: October 2023

The information contained herein is the confidential and proprietary information of ASG Technologies Group, Inc. Unauthorized use of this information and disclosure to third parties is expressly prohibited. This technical publication may not be reproduced in whole or in part, by any means, without the express written consent of ASG Technologies Group, Inc.

© 1982-2023 ASG Technologies. All rights reserved.
All names and products contained herein are the trademarks or registered trademarks of their respective holders.

ASG Technologies Worldwide Headquarters Naples Florida USA | asg.com | info@asg.com
708 Goodlette Road North, Naples, Florida 34102 USA Tel: 239.435.2200 Fax: 239.263.3692 Toll Free: 800.932.5536 (USA only)

Contents

Preface . . . . . **vii**

About this Publication . . . . . **vii**

E-mail User Forum . . . . . **viii**

Related Publications . . . . . **ix**

Publication Conventions . . . . . **x**

Worldwide Customer Support . . . . . **xi**
Intelligent Support Portal (ISP) xi
Product Support Policy xii

ASG Documentation/Product Enhancements . . . . . **xii**

Chapter 1: ZEKE Batch Utility **1**

Batch Utility Commands **2**
Statement Format 3
Input Source 6
Command Acceptance 6
Program JCL 6

BACKUP—Copy the Database or Audit Dataset **8**
Database Backup Size Report 10

CALENDAR—Add/Maintain System Calendars **14**

CLEARCPU—Clear the Database CPU Record **19**

CREATE—Initialize the Database **20**

CREDENTIAL—Manage Zeke Credentials **22**

DOC—Copy Documentation into the Database **23**

EVENT—Add/Maintain Event Master Records **29**

GENOPTS—Maintain Generation Options **68**

JCL—Copy JCL into the Database **70**

LOGIN—Manage Zeke Logins **72**

OPTIONS—Perform Administrative Tasks **78**

RECOVER—Restore Events from Backup **83**

RESOURCE—Maintain Resources **85**

RESTORE—Restore a Database Backup. 87

SCHEDULE—Create the New Schedule 93
Simultaneous Schedule Load (Simuload) 94
OVERRIDE—Select Events 102
REPORT—Generate Schedule Reports 104

SIMULATE—Simulating the Schedule 106
COPY—Create a Simulation Database 110
REPORT—Generate Simulation Reports 111

VAULT—Control the Vault Dataset. 113

Chapter 2: ZEKEXUTL Import/Export Utility. 115

Importing and Exporting Database Records 116

Implementation Requirements 116
Allocate XML Datasets 116
Install RI Server 119
Configure the Zeke Server 119
Define OMVS Segment in User Profile 120

Executing the ZEKEXUTL Utility 120
LE Runtime Options 122
Log Messages 122
Control Statements 122

EXPORT—Export Database Records to a File 127
EXPORT Examples 133

IMPORT—Import Records from a File or Database 134
IMPORT Example 137

FILTER—Filter Imported/Exported Records. 138
Filtering Rules 141

CHANGE—Change Imported Record Values. 144
CHANGE Examples 149
Multiple Change Rule Processing 150

REPORT—Creating Import/Export Reports 152
Report Examples 154

Sample Import and Export Jobs 160
Sample EMR Export Jobs 160
Sample Calendar Export Job 162
Sample Calendar Import Job 163
Sample Mass Record Change Jobs 164
Moving from a Test to a Production System 166

Chapter 3: ZEKESET Utility 169

Using ZEKESET 170

SET Statement 170
SET ABEND—Terminate a Job 175
SET ROUTE—Create Large Block Letter Separator Pages 175
SET xCOM—Issue System and Zeke Commands 176
SET VARIABLE—Define Zeke Variables 179
SET COMMENT—Add Comment Lines 181
SET CONDCODE/RETCODE—Set Condition/Return Codes 181
SET GOTO—Branch within Control Statements 183
SET WAIT—Set a Wait Interval 184
OPTION—Set ZEKESET Execution Options 185
CDATE—Perform Date Calculations 186
ON ERROR—Act on Syntax Errors 195

Chapter 4: Report Writer 199

Generating Reports 200
Output Reports to File 203

Report Formatting 204
Report Sequence—Event/Plan Listings 204
Page Breaks—Event/Plan Listings 205
Line Spacing—Event/Plan Listings 206
Report Output—Event/Plan Listings 207
Selecting Fields—Event/Plan Listings 208

LIST EVENTS/PLAN—EMR/SQR Listings 213

LIST CALID—Zeke Calendar Information 234

LIST CREDENTIAL—Zeke Credentials 237

LIST GENOPTS—Zeke Generation Options 239

LIST LOGIN—Zeke Logins 241

LIST OPERCLAS—Operator ID Security Classes 243

LIST OPERRECS—Zeke Operator ID Information 245

LIST RESOURCE—System Resources 247

LIST VARIABLE—Zeke Variable Information 249

Chapter 5: Operator Commands 253

Issuing Zeke Operator Commands 255
Command Prefix 255
Multiple Commands 256
Text Conventions 256
Command Abbreviations 256
Command Syntax 257
Command Acceptance 260
Command Authentication 260

Global Parameters 261

ZADD Command 274

ZALTER Command 286
Altering JCL Source 286
Altering Other Event Fields 287
Altering Message Generation 297
Altering Initiator Availability 298

ZDELETE Command 300

ZDISABLE Command 308
Disabling Scheduled Events 308
Disabling Automatic Replies 309
Disabling Electronic Vaulting 310

ZDISPLAY Command 311
ZDISPLAY Formats 311
Displaying Event Information 311
Displaying Preceding and Succeeding Events (PathFinder) 332
Displaying Variables 335
Displaying Initiator Information 336
Displaying Automatic Reply Information 340
Displaying System Pool Information 341
Displaying Zeke Agents 341
Displaying Remote Agents 343
Displaying Remote Events, Variables and Triggers 345
Displaying Communication Record Information 348
Displaying Database Information 348
Displaying Generation Options 349
Displaying Tracing Messages 351

ZENABLE Command 360

ZHOLD Command 361

ZID Command 365

ZINFO Command 366

ZKILL Command 367

ZMAP Command 370

ZOK Command 372

ZPLEX Command 373

ZREFRESH Command 383

ZRELEASE Command 384

ZRELOAD Command 388

ZRESOURCE Command 391

ZSCAN Command 395

ZSET Command 397
Setting Variable Values. 397
Setting the Default System ID. 399

ZSTATUS Command 400

Zeke Address Space Commands 401
#APPEND 402
#CLEAR 403
#DISPLAY 404
#RESET. 405
#START 407

Zeke Server Commands 408
$CLOSE 408
$DISPLAY 410
$KILL 416
$OPEN 419
$SET 420

Chapter 6: Field Descriptions 425

Chapter 7: Generation Options 519

Appendix A: XPath Location Paths 609

General Format 609

Event (EMR) Field Names 612

Variable (VAR) Field Names 617

Calendar (CAL) Field Names 617

Login Field Names. 618

Expressions 620

Appendix B: LE Runtime Options 623

Language Environment (LE) Overview 623

Overriding the Program Runtime Options. 623
Zeke Server (ZEKE6SRV) 624
Import/Export Utility (ZEKEXUTL) 624
Zeke Address Space Commands. 625
Tuning Notes and Precautions 626

Appendix C: Other Address Space Commands 629

Enabling the Commands 629
Command Format 630
#ABEND Command 630
#RESTART Command 632
#STOP Command 633

Index 635

Preface

This *ASG-Zeke Scheduling for z/OS Reference Guide* provides information about the enterprise scheduler ASG-Zeke Scheduling (herein called Zeke). This guide assumes that the appropriate components have been installed at your site.

This publication may refer to these additional, related ASG products:

- ASG-OASIS (herein called OASIS)
- ASG-OASIS/DMS (herein called OASIS/DMS)
- ASG-OpsCentral (herein called OpsCentral)
- ASG-RI Server (herein called RI Server)
- ASG-Zack (herein called Zack)
- ASG-Zara (herein called Zara)
- ASG-Zebb (herein called Zebb)
- ASG-Zeke Agent (herein called Zeke Agent)
- ASG-Zena (herein called Zena)

About this Publication

This publication consists of these chapters:

- Chapter 1, "ZEKE Batch Utility," explains the ZEKE batch utility program functions and parameters.
- Chapter 2, "ZEKEXUTL Import/Export Utility,"explains the import/export batch utility program ZEKEXUTL.
- Chapter 3, "ZEKESET Utility," provides information on using Zeke job and input control statements.
- Chapter 4, "Report Writer," describes the reporting function of the ZEKE batch utility program.
- Chapter 5, "Operator Commands," describes the operator commands used to make changes from any operating system console or via the /COMM or /ZCOM functions of the Zeke online system. Most commands can also be issued from any command line in the Zeke ISPF online facility.

- Chapter 6, "Field Descriptions," alphabetically lists and explains all input fields in the Zeke online facility.
- Chapter 7, "Generation Options," alphabetically lists and explains all Zeke generation options.

E-mail User Forum

To share information, ask questions, receive tips, or compare notes, join the Zeke e-mail user group, *Autoops*. It is easy to subscribe (and to unsubscribe, if necessary).

To subscribe to Autoops

1. Visit the Autoops Home Page at: http://list.asg.com/scripts/wa.exe?A0=AUTOOPS
2. From the Options, click the link Subscribe or Unsubscribe.
3. Complete the form:
   a. Enter your name and e-mail address.
   b. Select the Autoops list.
   c. Select your preferred subscription type.
   d. Click Subscribe (Autoops).

After your request is received, you will receive an e-mail confirming your membership, and providing additional instructions on participating in the group and managing your subscription.

As a member, you will receive copies of all new messages sent by other members of the group. From the Autoops Home Page, you also can search an archive of past messages.

To submit a question or comment to the group

- Send an e-mail to the group members at: autoops@list.asg.com

To manage your subscription

- Log in to the Subscriber's Corner at: http://list.asg.com/scripts/wa.exe?SUBED1=AUTOOPS

To contact the list owner

- Send an e-mail to the list owner at: autoops-request@list.asg.com

Related Publications

The documentation library for ASG-Zeke Scheduling for z/OS consists of these publications (where *nn* represents the product version number):

- *ASG-Zeke Scheduling for z/OS Enhancement Summary* (AZM1000-*nn*) describes new Zeke features, updated functions, and performance improvements.
- *ASG-Zeke Scheduling for z/OS User's Guide* (AZM0200-*nn*) explains the procedures for using Zeke for enterprise scheduling.
- *ASG-Zeke Scheduling for z/OS Installation Guide* (AZM0300-*nn*) defines Zeke system requirements, provides instructions for installing Zeke, and explains the optional features you can activate after installing Zeke.
- *ASG-Zeke Scheduling for z/OS Reference Guide* (AZM0400-*nn*) provides a reference for using Zeke batch programs and operator commands, and for generating reports.
- *ASG-Zeke Scheduling Messages and Codes Guide* (AZM1200-*nn*) lists the Zeke messages, describes their causes and resolutions, and provides return code explanations.
- *ASG-OASIS for z/OS Enhancement Summary* (AZO1000-*nn*) describes new OASIS features, updated functions and performance improvements.
- *ASG-OASIS for z/OS Installation Guide* (AZO0300-*nn*) provides information on installing ASG-OASIS, the framework for your ASG enterprise workload management ('Z') products.
- *ASG-OASIS for z/OS Reference Guide* (AZO0400-*nn*) provides information on OASIS commands, options, and other functions.
- *ASG-OASIS Messages and Codes Guide* (AZO1200-*nn*) lists the OASIS messages, describes their causes and resolutions, and provides return code explanations.

Note:

To obtain a specific version of a publication, contact ASG Customer Support.

Publication Conventions

ASG Technologies uses these conventions in technical publications:

Convention	Usage
Arrow (▸)	Used in a procedure to indicate commands within menus. Also used to denote a one-step procedure.
Bold	Indicates that case-sensitive usage is required for a directory, path, file, dataset, member, database, program, command, or parameter name. ▸ Verify the settings in the **asg.conf** file.
Capitalization	For system element names, this varies according to the product interface and its operating environment. Mainframe file names use uppercase, for example: ▸ Allocate a JSOPTEM member in the JLRCL library. Windows file names use mixed case, for example: ▸ Create a text file named SECLIST.txt in the C:\Program Files\ASG\config directory. UNIX file names use mixed case, for example: ▸ Edit the **databaseID.ACC** file in the **/database** directory. Typical product and operating system elements include: • Directory, path, file, dataset, member, database, program, command, and parameter names. • Window, field, field group, check box, button, panel (or screen), and option labels. • Names of keys. A plus sign (+) is inserted for key combinations (e.g., Alt+Tab).
`lowercase italic monospace`	Information that you provide according to your particular situation. For example, you would replace *`filename`* with the actual name of the file.
`Monospace`	Characters you must type exactly as they are shown, such as code, JCL, file listings, or command/statement syntax. Also used for denoting brief examples in a paragraph.
Underline	Denotes a cursor-selectable field or line.

Convention	Usage
Vertical separator bar (|) with underline	Indicates options available with the default value underlined (e.g., Y|N).

Worldwide Customer Support

ASG Technologies provides support throughout the world to resolve questions or problems regarding installation, operation, or use of our products. ASG provides all levels of support during normal business hours and emergency support during non-business hours.

You can access support information from ASG's Support page.

ASG Third-party Support. ASG provides software products that run in a number of third-party vendor environments. Support for all non-ASG products is the responsibility of the respective vendor. In the event a vendor discontinues support for a hardware and/or software product, ASG cannot be held responsible for problems arising from the use of that unsupported version.

Intelligent Support Portal (ISP)

The ASG Intelligent Support Portal (ISP) provides online support.

Log on to the ISP with this information:

Customer ID = *NNNNNNNNN*

Password = *XXXXXXXXXX*

where:

NNNNNNNNN is your customer ID supplied by ASG Product Distribution.

XXXXXXXXXX is your unique password supplied by ASG Product Distribution.

If you do not have your logon information, contact your local support center.

This table outlines the support response times you can expect:

Severity	Meaning	Expected Support Response Time
1	Production down, critical situation	Within 30 minutes
2	Major component of product disabled	Within 2 hours
3	Problem with the product, but customer has work-around solution	Within 4 hours
4	"How-to" questions and enhancement requests	Within 4 hours

Product Support Policy

ASG Technologies fully supports the current release and one previous release of each of its products. ASG will temporarily support an older release, for up to six months, to provide time for you to upgrade.

Once programming support for a product release is withdrawn, ASG will no longer supply new fixes for problems nor accept enhancement requests for that release. When a vendor announces the end of support for system software or a hardware configuration on which ASG products rely, ASG will make a similar announcement regarding the support plans for its products. ASG's support for problems affected by system software release levels will terminate when the vendor no longer supports their hardware or software. Announcements regarding support plans for various products can be found on ASG's Web site.

ASG Documentation/Product Enhancements

Use ASG's contact form to submit your product and documentation suggestions. Ensure that you include the name of the ASG Technologies product you are using. For documentation suggestions, include the publication number located on the publication's front cover.

1 ZEKE Batch Utility

This chapter describes the ZEKE batch utility program, the format of the program statements, and the commands and keywords. It also includes a description of the Simulation program. This chapter contains these topics:

Topic	Page
Batch Utility Commands	2
Statement Format	3
Input Source	6
Command Acceptance	6
Program JCL	6
BACKUP—Copy the Database or Audit Dataset	8
Database Backup Size Report	10
CALENDAR—Add/Maintain System Calendars	14
CLEARCPU—Clear the Database CPU Record	19
CREATE—Initialize the Database	20
CREDENTIAL—Manage Zeke Credentials	22
DOC—Copy Documentation into the Database	23
EVENT—Add/Maintain Event Master Records	29
GENOPTS—Maintain Generation Options	68
JCL—Copy JCL into the Database	70
LOGIN—Manage Zeke Logins	72
OPTIONS—Perform Administrative Tasks	78
RECOVER—Restore Events from Backup	83
RESOURCE—Maintain Resources	85
RESTORE—Restore a Database Backup	87
SCHEDULE—Create the New Schedule	93
Simultaneous Schedule Load (Simuload)	94
OVERRIDE—Select Events	102
REPORT—Generate Schedule Reports	104
SIMULATE—Simulating the Schedule	106
COPY—Create a Simulation Database	110

Topic	Page
REPORT—Generate Simulation Reports	111
VAULT—Control the Vault Dataset	113

Batch Utility Commands

These are the key functions that you can perform using the ZEKE utility program:

Command	Description
BACKUP	Backs up the Zeke database.
CALENDAR	Adds and maintains Zeke calendars.
CLEARCPU	Deletes all entries from the Zeke database CPU record.
CREATE	Initializes a Zeke database.
CREDENTIAL	Manages Zeke credentials.
DOC	Copies documentation into the database.
EVENT	Adds and maintains EMRs.
GENOPTS	Adds a new local GENOPT, deletes a GENOPT from the Zeke database, or updates specific field values for a particular GENOPT.
JCL	Copies JCL into the Zeke database.
LOGIN	Manages Zeke logins.
OPTIONS	Performs a variety of functions (e.g., updating your customer ID or generating a Database Status Report).
RECOVER	Enables you to restore individual EMRs from a tape or disk backup file that was created by the BACKUP function.
RESOURCE	Enables you to add, update, and delete logical resources.
RESTORE	Restores the Zeke database from a backup.
SCHEDULE	Creates the daily schedule.
OVERRIDE	Includes or exclude events from the schedule regardless of their OCCURS clause.

Command	Description
REPORT	Generates schedule reports.
SIMULATE	Creates a simulation of the Zeke schedule and produces schedule reports.
VAULT	Controls the vault dataset

Statement Format

These are the formatting rules for ZEKE utility program input statements:

- Begin a statement in any column and end the statement by column 72. Columns 73 through 80 are ignored.
- Continue on as many statements as necessary in any column (from 1 through 72). No continuation character is required. All parameters following a command are assumed to belong to that command until the next command word is encountered.
- Optional. Enter `$$` (two dollar signs) to signify the end of a statement, so Zeke knows to process the statement without reading the next command first.
- Separate parameters and operands by either an equal sign (=) or by one or more spaces.
- Enclose multiple operands within parentheses and separate by a comma or by one or more spaces. Parentheses are optional for one operand.
- Separate parameters by a comma or by one or more spaces. Use either a comma or one or more spaces to separate items, but not both. A comma followed by a space indicates the end of the statement to Zeke.
- If you are running multiple versions of Zeke, the EXEC statement must include the subsystem name (if other than SSSI). For example:

```
//Z1    EXEC  ZEKEUTL,PARM='SUBSYS=subsystem name'
```

Character String Delimiters

Use a special character to delimit character string operands. Special characters include any HEX character of 7F or less (e.g., a quote `(')`, dash `(-)`, slash `(/)`, or parentheses `[( )]`.

Note:

These characters are exceptions and cannot be used as delimiters:

— dollar sign ($)

— question mark (?)

— number/pound sign (#)

— at sign (@)

— asterisk (*)

The delimiter character cannot appear within the character string.

These are valid examples of the use of character string delimiters:

```
EVENT UPD 3 DESC='THIS IS THE DESCRIPTION'
EVENT UPD 45 DESC .THE DESCRIPTION IS DELIMITED BY A PERIOD.
EVENT UPD 53 DESC=/SLASHES OR OTHER CHARACTERS MAY BE USED/
EVENT UPD 77 DESC=(PARENTHESES MAY BE USED FOR DELIMITERS)
```

These are *invalid* examples of the use of character string delimiters:

```
EVENT UPD 4 DESC=(AN OPEN PAREN MUST BE FOLLOWED BY A CLOSE(
EVENT UPD 34 DESC 'DELIMITER CHAR 'QUOTE' CONTAINED IN VALUE'
EVENT UPD 54 DESC=$DOLLAR SIGN MAY NOT BE USED AS DELIMITER$
```

Delimiters indicate that a value is character instead of numeric. When there are no delimiters and the value is all numbers, the value is numeric; otherwise, it is character.

When there are no delimiters, a character value can be only one word with no spaces, commas, or equal signs.

These are examples of character, character string, and numeric values:

```
WHEN (VAR $XYZ LE 55)                 Numeric value
WHEN (VAR $XYZ LE '55')               Character value
WHEN (VAR $XYZ EQ YES)                Character value
WHEN (VAR $XYZ LE 'YES')              Character value
WHEN (VAR $XYZ LE /CHAR STRING/)      Character string
```

OCCURS and WHEN Clauses

Enclose the entire OCCURS and/or WHEN clause within a set of parentheses; however, the clause can contain internal levels of parentheses. For example:

```
OCCURS (WORKDAYS AND (DAY GE 8 AND DAY LE 14))
WHEN (EOJ JOBNAME1)
```

Verify that the clauses contain as many right parentheses as left parentheses.

See the *ASG-Zeke Scheduling for z/OS User's Guide* for full instructions on how to create OCCURS and WHEN clauses.

Jobnames

Batch commands support the use of 30-character mixed-case jobnames, with the exception of keywords (which still must be entered in upper case).

You must change the editor to CAPS OFF before entering mixed-case jobnames. In CAPS OFF mode, MVS jobnames are *not* made upper case automatically. You must enter them in upper case.

This statement is valid:

```
EVENT ADD JOB PayrollReports
```

This statement is *invalid*:

```
event add job PayrollReports
```

With the WHEN condition, you can enter a 30-byte mixed-case jobname as the target of the jobname triggers (e.g., EOJ, AEOJ, and NOTDURING JOB). Within the WHEN condition, Zeke considers case in the jobnames only, not in other WHEN keywords. Both of these examples are valid WHEN conditions:

```
WHEN EOJ JOBA AND EOJ PayrollReports
WHEN eoj JOBA and eoj PayrollReports
```

Generic Selection Criteria

When using a Zeke batch function that enables you to select events in the Zeke database, you can use wildcard or placeholder characters in your selection criteria. These are the most common event attributes that enable the use of wildcards and placeholders in operands entered for these parameters:

- Application ID
- Event name
- Group ID
- System ID
- User ID
- Jobname

You can use an asterisk (*) as a wildcard for any number of characters. A wildcard functions in these ways:

- An asterisk at the end of an operand string (e.g., ABC*), selects any name (of any valid length) that *begins* with the specified characters.
- An asterisk at the beginning of an operand string (e.g., *ABC), selects any name (of any valid length) that *ends* with the specified characters.

- An asterisk in the middle of an operand string performs a wildcard search for any name matching the specified beginning and ending characters (plus any characters in between).

You can use a question mark (?) as a placeholder for any single character.

You can use wildcards and placeholders in combination.

Input Source

The ZEKE utility program reads input using the ddname SYSIN or from the console. To read input from the console, specify the CONSOLE parameter on the EXEC statement in the JCL or include the CONSOLE command as SYSIN data. The CONSOLE command must not be the last Zeke command supplied through the SYSIN; otherwise, the command is ignored.

Command Acceptance

A message is printed on the SYSPRINT dataset describing the results of the ZEKE utility program request. The last character of the message ID indicates these conditions:

Last Character	Description
I	The completion of a normal system function. For information only.
E	A severe error. The requested action is not completed.
W	A potential error. The requested action is completed.
R	An operator response is required.

Program JCL

The Zeke SZEKINS0 library contains the sample procedure named ZEKEUTL for executing the ZEKE utility program. The procedure contains the necessary DD statements for ZEKECAT, SYSPRINT, and SORTWK1 through SORTWK3.

Note:

The ZEKECAT DD is used only if the ZEKE utility program and the Zeke started task run on the same system (and use the same OASIS subsystem), but the Zeke started task is not running currently. Otherwise (if the Zeke started task is running), the ZEKE utility program and the Zeke started task use the same Zeke database.

Examples

```
//ZUTIL JOB ,MSGLEVEL=(1,1) ,CLASS=A
//Z1    EXEC  ZEKEUTL,PARM='SUBSYS=subsystem'
//SYSIN DD *
  EV ADD JOB ....
  EV ADD JOB ....
  EV ADD JOB ....
/*
//
```

```
//ZUTIL JOB ,MSGLEVEL=(1,1) ,CLASS=A
//Z1    EXEC  ZEKEUTL,PARM='SUBSYS=subsystem'
//SYSIN DD *
  SCHEDULE TODAY ACTIVATE
/*
//
```

```
//ZUTIL JOB ,MSGLEVEL=(1,1) ,CLASS=A
//Z1    EXEC  ZEKEUTL,PARM='SUBSYS=subsystem'
//SYSIN DD *
  LIST EVENTS fields=(event,ename,system)
/*
//
```

BACKUP—Copy the Database or Audit Dataset

The BACKUP function copies the contents of the Zeke database or the audit dataset to a tape or disk file (which serves as a backup copy in case the database must be restored). You use the RESTORE function to restore a database backup (see "RESTORE—Restore a Database Backup" on page 87).

Use the BACKUP function to back up your database at least once daily. ASG recommends that you back up the database prior to each new schedule run.

The database is copied in these two formats:

Format	Description
Physical	The database copied to tape is an exact copy of the database on disk.
Logical	The database copy is a reorganized, logical copy (event numbers are unchanged) with pointers to the various types of records. Event elements are grouped together.

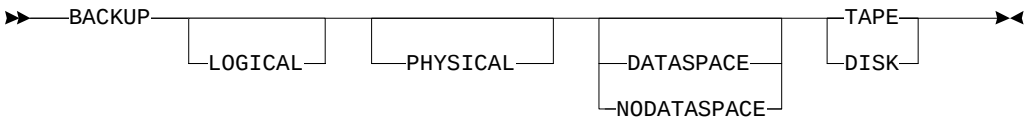
Caution! The Zeke database is not an ordinary sequential file. Most third-party backup/copy utilities will not back up the Zeke database successfully. Be sure to use only the ZEKE utility program's BACKUP and RESTORE functions for this purpose.

The Zeke database BACKUP ddname is ZEKEBK. In the ZEKEUTL jobstream, specify the Zeke backup file dataset name (this is a sequential file).

When you back up the database, a database backup report is generated automatically (see "Database Backup Size Report" on page 10).

See the *ASG-Zeke Scheduling for z/OS User's Guide* for more information on backing up the Zeke database.

Syntax



Parameters

You can use these parameters with the BACKUP function:

Parameter	Description
blank	*Recommended.* Creates both a physical and logical database backup. You can use the backup for a physical or logical restore.
LOGICAL	Because a logical backup always is performed (unless the PHYSICAL parameter is specified), this parameter no longer is effective, but is supported for backwards JCL compatibility.
PHYSICAL	Creates *only* an exact copy of the Zeke database and produces a full-track read and write of the physical portion of the database. You cannot use the backup file to restore the database logically, but you can use it with the RESTORE PHYSICAL parameter to move the database physically to another dataset. ASG recommends that you *do not use* this parameter because only a physical backup is performed (which limits restore to a physical restore). Instead, specify no parameters to create both a physical and logical backup.
DATASPACE	Optional. Creates a backup copy of the Zeke database (i.e., physical, logical, or both) from a temporary copy of the database created in a data space. Using this parameter reduces the I/O against the database, the duration of the database enqueue, and the time needed for the backup. **Note:** If you always want to use a data space for backups, set the DSPBatch generation option to Y (see page 541).
NODATASPACE	Optional. If the DSPBatch generation option is set to Y (see page 541), then the ZEKE utility program uses a data space when backing up the database. Use this parameter to override this for a particular backup so that a data space is not used.
TAPE	Default. Copies the backup to a tape. If no other parameters are specified, this option creates a physical backup and a logical backup of the Zeke database and copies the backups to a tape.
DISK	Copies the backup to a disk file. If no other parameters are specified, this option creates a physical backup and a logical backup of the Zeke database and copies the backups to a disk file.

Examples

This is an example of the JCL for the BACKUP function to back up the Zeke database to tape:

```
//ZEKEBKUP  JOB  ,MSGLEVEL=(1,1),CLASS=A
//ZBK       EXEC ZEKEUTL
//ZEKEBK    DD   DSN=ZEKE.BACKUP,DISP=(NEW,KEEP),
//           VOL=(RETAIN,SER=ZEKETP),UNIT=TAPE,LABEL=(1,SL)
//SYSIN     DD   *
  BACKUP TAPE DATASPACE
/*
```

This is an example of the JCL for the BACKUP function to back up the Zeke database to disk:

```
//ZEKEBKUP  JOB  ,MSGLEVEL=(1,1),CLASS=A
//ZBK       EXEC ZEKEUTL
//ZEKEBK    DD   DSN=ZEKE.DISK.BACKUP,DISP=(,KEEP)),
//           UNIT=SYSDA,SPACE=(CYL,(10,10)),
//           VOL=SER=ZEKEVL
//SYSIN     DD *
  BACKUP DISK DATASPACE
/*
```

The Zeke database is enqueued for the duration of the physical backup unless the DATASPACE parameter is used, or the DSPBatch generation option is set to Y (see page 541). ASG recommends that you schedule the backup during the period that has the least amount of activity if you are not using a data space.

Database Backup Size Report

The BACKUP function produces a report to help you determine the optimum block size and number of cylinders for the Zeke database.

ASG makes these recommendations for determining your optimum database size:

- Run Zeke for at least one week before you make any modifications to the database based on the information in the database backup report.
- For a new database, use the default block size of 2048.

  See the section on database allocation in the *ASG-Zeke Scheduling for z/OS Installation Guide* for information to help you determine the number of cylinders required for a new database.
- Allocate enough space so that your database is 50 percent full. The PctWarn generation option (see page 564) controls how full the database can become before a warning message is issued. The default is 80 percent.

Figure 1 illustrates an example of the database backup size report.

Figure 1 • Database Backup Size Report

```
EVENTS BY SIZE RANGE
    0 - 1024:  73,191                  Smallest event size:     420
 1025 - 1536:  29,071                   Largest event size:  19,748
 1537 - 2048:  13,059                   Average event size:   1,091
 2049 - 2560:   2,276                 Highest event number: 119,031
 2561 - 3072:     656
 3073 - 3584:     320
 3585 - 4096:     153
   over 4096:     268

 SPACE NEEDED FOR EVENTS BY BLOCK SIZE            TOTAL SPACE NEEDED BY BLOCK SIZE

Block     Blocks      Tracks  Cylinders             Blocks      Tracks  Cylinders
 Size     Needed      Needed     Needed             Needed      Needed     Needed

 1024    169,829       5,147        344            187,501       5,682        379
 1536    136,729       5,259        351            148,233       5,702        381
 2048    123,002       5,858        391            131,727       6,273        419
 2560    120,538       7,091        473            127,461       7,498        500
 3072    119,786       7,986        533            125,611       8,375        559
 3584    119,449       9,189        613            124,437       9,573        639
 4096    119,278       9,940        663            123,671      10,306        688
```

This table describes the fields that are displayed on the report:

Field	Description	
EVENTS BY SIZE RANGE	Displays the number of events that will fit in each possible block size. In the example, most of the events will fit into a 2048-byte block (with relatively few events over 2048 bytes in size). These additional event statistics are displayed:	
	Smallest event size	Displays the size of the smallest event.
	Largest event size	Displays the size of the largest event. In the example, the event indicated as largest is an SCOM event with numerous commands.
	Average event size	Displays the average event size.

Field	Description	
	Highest event number	Displays the highest event number in the database. There must be sufficient event capacity to accommodate the highest event number when a database restore is done (even if there are many empty events below the highest event number.) By default, the event capacity (i.e., highest event number) is calculated as 75 percent of the number of blocks in the database. You can override this value using the MAXEVENTS parameter with the CREATE or RESTORE command.
SPACE NEEDED FOR EVENTS BY BLOCK SIZE	Displays the number of blocks, tracks, and cylinders that are required (for events only) to contain the backed-up database for each possible block size.	
	Block Size	Displays the possible block sizes.
	Blocks Needed	Displays the number of blocks required.
	Tracks Needed	Displays the number of tracks required.
	Cylinders Needed	Displays the number of cylinders required.
TOTAL SPACE NEEDED BY BLOCK SIZE	Displays the number of blocks, tracks, and cylinders that are required (for all records—e.g., events, variables, resources, etc.) to contain the backed-up database for each possible block size.	
	Blocks Needed	Displays the number of blocks required.
	Tracks Needed	Displays the number of tracks required.
	Cylinders Needed	Displays the number cylinders required.

Optimum block size for the Zeke database balances number of blocks and cylinders and enables each event to fit into one block (in most cases). As the number of blocks needed decreases, the number of cylinders increases (as the block size increases).

The more cylinders that a database occupies, the longer it takes to perform operations that require processing the entire database (e.g., creating a data space and backing up the database).

To determine optimum database size using the report

1 Determine the recommended block size for the new allocation.

Based on the example in Figure 1 on page 11, the recommended block size is 2048. Because most events fit into a 2048-byte block, significantly fewer cylinders (81) are required than would be required for a block size of 2560.

2 Calculate the total database size (based on the recommended block size) so that the database is used at 50 percent of total capacity. For example, if a total capacity of 419 cylinders is needed to accommodate existing data, then you should allocate 838 cylinders for the new database:

```
419 × 2 = 838
```

Calculate the resulting number of blocks (see the database allocation table in the *ASG-Zeke Scheduling for z/OS Installation Guide*). For example, the resulting 838 cylinders (with a block size of 2048) yield 263,970 blocks:

```
838 × 315 = 263,970
```

3 *z/OS*Determine the resulting event capacity. For example, 75 percent of the resulting 263,970 blocks yields a default event capacity of 196,657 blocks. This enables an additional 77,626 events to be added to the database (assuming there no empty events in the database already).

To use the new size values

1 Re-cycle all Zekes (using ZKILL COLD or TRACK) that share the database.

2 Back up the existing Zeke database.

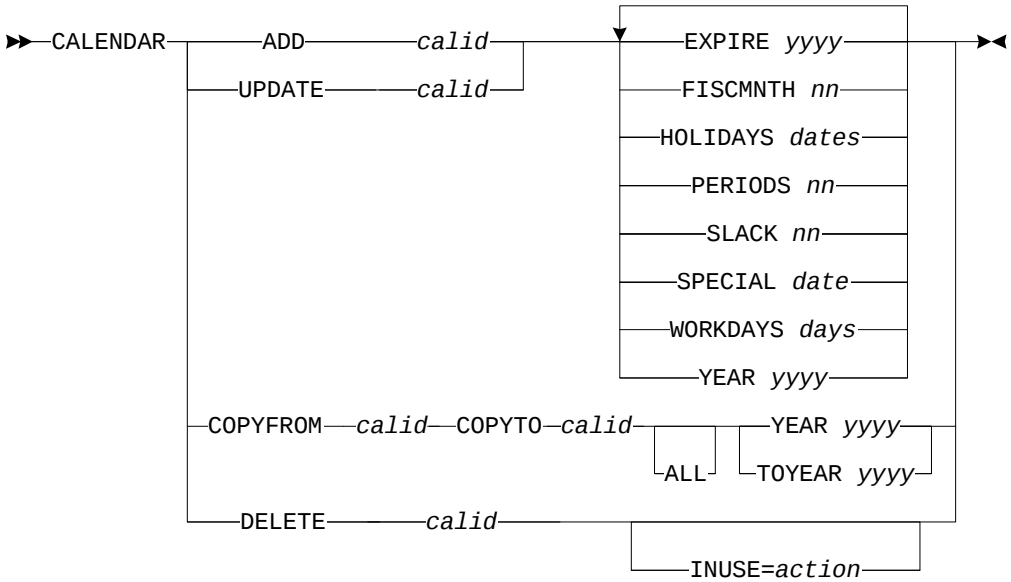
3 Allocate the new Zeke database.

4 Perform a logical restore from the backup into the new database.

CALENDAR—Add/Maintain System Calendars

The CALENDAR function adds, copies, updates, or deletes a Zeke calendar.

Syntax



Parameters

You can use these parameters with the CALENDAR function:

Parameter	Description
These are the valid actions for the CALENDAR function:	
ADD	Adds a new calendar. For example: `CALENDAR ADD TEST1 FISCMNTH 10` **Note:** You can define multiple calendars in the Zeke database, which could have different working days and different holidays.
COPYFROM	Copies the specified existing calendar. For example, this command creates a new calendar named STDCAL2 for 2020 from a copy of STDCAL1 for 2020. `CALENDAR COPYFROM STDCAL1 YEAR 2020 COPYTO STDCAL2`

Parameter	Description
`DELETE`	Deletes the existing calendar from the Zeke database. **Note:** You can include the INUSE keyword (see page 16) with the DELETE parameter to indicate the action to take if Zeke determines that the calendar is in use. (If you delete a calendar that is specified for an active EMR, Zeke may not be able to schedule the event.)
`UPDATE`	Updates the workdays and/or holidays of an existing calendar.

These are the valid parameters for the CALENDAR function:

`ALL`	Indicates whether to copy all documentation records from the existing calendar to the new calendar.
`calid`	Specifies the name (up to eight characters long) of the calendar.
`COPYTO`	Specifies the name of the calendar to be created by the COPYFROM command. For example, this command creates a new calendar named B for 2020 from a copy of existing calendar A and also copies all existing documentation: `CALENDAR COPYFROM A COPYTO B TOYEAR 2020 ALL`
`date`	Specifies the date that the calendar expires (in *`mm/dd/yyyy`* or *`dd/mm/yyyy`* format, depending on your system's date setting). **Note:** You can update the expiration date using the CALENDAR function of the ZEKE utility program. To delete an expiration date, you must use the Zeke online facility. See the *ASG-Zeke Scheduling for z/OS User's Guide* for more information on Zeke calendars.
`FISCMNTH` *`nn`*	Required with the ADD function. Specifies the starting fiscal month for a standard calendar. The valid values range from `1` through `12` (i.e., January through December).

Parameter	Description
HOLIDAYS (*dates*)	Specifies up to 30 holiday dates for a standard or user accounting calendar (in either *mm/dd/yyyy* or *dd/mm/yyyy* format, depending on your system's date setting). Enclose multiple dates in parentheses and separate them by commas or spaces. **Note:** When you update holidays, enter all holidays that you want to remain in effect on the statement; any dates that you exclude from the statement are considered normal working days.
INUSE	Valid with the DELETE keyword. Specifies the action to take if Zeke determines that the calendar is in use by an event. These are the valid values: FAIL Default. Issues a warning message and terminates the delete request. IGNORE Disables the in-use check entirely. No message is issued. WARN Issues a warning message and allows the delete request.
PERIODS *nn*	Specifies the number of days in each period in a user accounting calendar. You can specify up to 24 periods. Enclose multiple dates in parentheses and separate them by commas or spaces.
SLACK *nn*	Specifies the number of slack days between periods in a user accounting calendar. The valid values range from 0 through 40.
SPECIAL *date*	Specifies the dates to indicate as scheduling dates in a special calendar (in either *mm/dd/yyyy* or *dd/mm/yyyy* format, depending on your system's date setting). You can specify up to 366 dates. The value for *yyyy* must match the calendar year. Enclose multiple dates in parentheses and separate them by commas or spaces.
TOYEAR *yyyy*	With the COPYTO function, this parameter specifies the year of the new calendar to be created from an existing calendar. If omitted, the new calendar is created for the same year as the COPYFROM calendar.
WORKDAYS *days*	Specifies all of the days that are defined as workdays for a standard or user accounting calendar. For example: MONDAY TUESDAY WEDNESDAY THURSDAY FRIDAY SATURDAY SUNDAY

Parameter	Description
	Note: When WORKDAYS are updated, enter all workdays on the statement. Any dates excluded from the list are considered non-working days.
YEAR *yyyy*	Required with the COPYFROM function. Specifies the year of the existing calendar to be copied. Required with the UPDATE or DELETE function for a year-specific calendar. Specifies the year that the calendar is valid. If left blank, the default value is `****`.

Examples

This example updates the standard calendar A with the appropriate holidays:

```
CALENDAR UPDATE A
HOLIDAYS (01/01/2020,07/04/2020,11/27/2020,11/28/2020,12/25/2020,01/01/2020)
FISCMNTH 1
```

This example adds the standard calendar B:

```
CALENDAR ADD B
WORKDAYS MONDAY TUESDAY WEDNESDAY THURSDAY FRIDAY SATURDAY
HOLIDAYS (01/01/2020,07/04/2020,11/27/2020,11/28/2020,12/25/2020,01/01/2020)
FISCMNTH 1
```

This example adds the user accounting calendar USER2:

```
CALENDAR ADD USER2 YEAR 2020 SLACK 02
WORKDAYS MONDAY TUESDAY WEDNESDAY THURSDAY FRIDAY SATURDAY SUNDAY
HOLIDAYS (01/01/2020,07/04/2020,11/27/2020,11/28/2020,12/25/2020)
FISCMNTH 1
PERIOD (28,28,32,28,28,32,28,28,32)
```

This example adds the special calendar SPECIAL4:

```
CALENDAR ADD SPECIAL4 YEAR 2020 EXPIRE 12/31/2020
SPECIAL (01/10/2020,02/10/2020,03/10/2020,04/10/2020,05/10/2020)
      FISCMNTH 1
```

CLEARCPU—Clear the Database CPU Record

The CLEARCPU function deletes *all* entries from the Zeke database CPU record.

The CPU record contains an entry for each Zeke system that is currently sharing the Zeke database. You can display these entries using the ZD COM command (see "ZDISPLAY Command" on page 311.)

If a Zeke system terminates abnormally, it cannot be removed from the CPU record.

Note:

You can delete an individual entry using the OPTIONS CPUDEL command. See "OPTIONS—Perform Administrative Tasks" on page 78.

Caution! Because the CLEARCPU command deletes *all* CPU record entries, ASG recommends that you use it with extreme caution. The command must be issued from a standalone batch job (i.e., a job running the ZEKE utility program while no other Zeke batch program or started task is accessing the Zeke database).

Syntax



CREATE—Initialize the Database

The CREATE function initializes a new or existing database or vault. When the CREATE function is performed on an existing database, all information in that database is destroyed. Typically, you use the CREATE function for one of these reasons:

- You are using Zeke for the first time.

  Note:

  Because the ZEKE utility program is independent of the operating system, it requires that OASIS be active. Special provisions have been made to enable OASIS to be activated without Zeke being active. This is a normal condition during the process of installing Zeke.

- You are using a new dataset.

See the *ASG-Zeke Scheduling for z/OS Installation Guide* for information on choosing the database size.

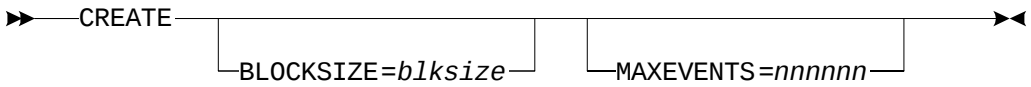
The CREATE function is not necessary if the database is being restored, because the RESTORE function automatically initializes the database (unless the MERGE parameter is specified; in this case, the CREATE functions are not performed).

Caution! Because the CREATE function does not check whether any Zeke systems currently are using the Zeke database to be created, ASG recommends that you use this function with caution when initializing an existing database.

The CREATE function uses the dataset name ZEKENEW. All other system functions use the dataset name ZEKECAT.

See the *ASG-Zeke Scheduling for z/OS User's Guide* for more information and additional considerations when creating the Zeke database and vault.

Syntax



Parameters

Parameter	Description
BLOCKSIZE=*blksize*	Optional. Specifies the block size of the Zeke database from 1024 through 4096 (in 512-byte increments). The default value is 2048. See "OPTIONS—Perform Administrative Tasks" on page 78 for information on the Database Status Report (which you can use to determine the optimum block size).
MAXEVENTS=*nnnnnn*	Optional. Specifies the maximum number of events (i.e., the highest event number) that can exist in the Zeke database. The valid values range from 1 through 999999. The default value is 75 percent of the number of blocks in the database. For example, a database allocated with 100 blocks can have a maximum of 75 events unless this parameter is used to override that value.

Sample JCL

This is an example of a database creation jobstream:

```
//ZEKECRET JOB    ,MSGLEVEL=(1,1),CLASS=A
//ZUTL     EXEC   ZEKEUTL
//ZEKENEW  DD     DSN=ZEKE.MVS.DATABASE,DISP=(NEW,CATLG),
//         UNIT=SYSDA,VOL=SER=ZEKEVL,SPACE=(CYL,(10))
//SYSIN    DD     *
  CREATE BLOCKSIZE=4096
/*
```

Note:

The Zeke database is a sequential file. Do not code a DSORG of any other type.

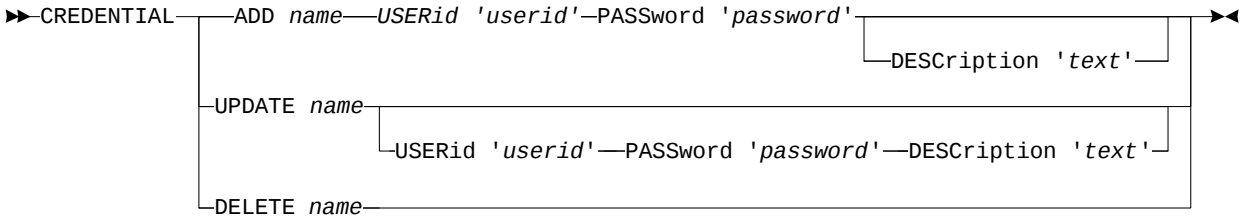
CREDENTIAL—Manage Zeke Credentials

The CREDENTIAL function manages Zeke credentials. Zeke stores credentials in the Zeke database so that they can be associated with login records for Zeke task events (i.e., tasks that Zeke submits to a remote Zena agent). Zeke requires the appropriate login and user credentials in order to gain access to a remote Zena server.

A Zeke *credential* supplies a valid user ID and password to be used for authentication. You create as many credential records as necessary, then you can map the credentials with the appropriate login records.

See the *ASG-Zeke Scheduling for z/OS User's Guide* for more information on Zeke credentials.

Syntax



Parameters

You can use these parameters with the CREDENTIAL function:

Parameter	Description
These are the valid actions for the CREDENTIAL function:	
`ADD`	Adds a new Zeke credential.
`DELETE`	Deletes an existing Zeke credential from the database.
`UPDATE`	Updates an existing Zeke credential.
These are the valid parameters for the CREDENTIAL function:	
`name`	Specifies the name (up to 32 characters long) of the credential. The name cannot be delimited.
`DESCription '`*`text`*`'`	Description of the credential. You can request to display this description when generating reports. The value must be enclosed in delimiters.

Parameter	Description
PASSword '*password*'	Required with the ADD function. Specifies the password associated with the credential's user ID. The value must be enclosed in delimiters.
USERid '*userid*'	Required with the ADD function. Specifies the user ID associated with the credential. The value must be enclosed in delimiters.

Sample JCL

This example illustrates the creation of a credential:

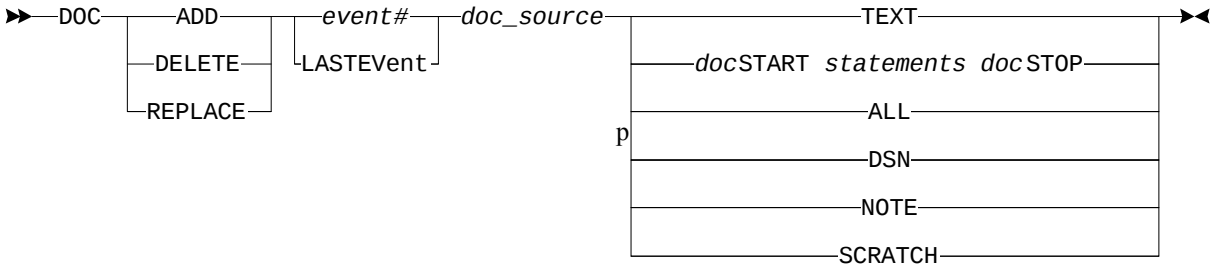
```
//ZEKECRED JOB  ,MSGLEVEL=(1,1),CLASS=A
//ZUTL     EXEC ZEKEUTL
//SYSIN    DD   *
 CREDENTIAL ADD SPAIN_FTP_USER
            USERID 'generalFTP'
            PASSWORD 'password'
            DESC 'General user for Spain FTP servers'
/*
```

DOC—Copy Documentation into the Database

The DOC function copies documentation from an outside source into the Zeke database.

This function requires a fixed block record length of 80.

Syntax



Parameters

You can use these parameters with the DOC function:

Parameter	Description
ADD	Adds documentation to an event from an outside source.
DELETE	Deletes the existing text documentation from an event.
REPLACE	Replaces an event's existing documentation with documentation from an outside source.
event#	Specifies the event number of the event to be updated.
LASTEVent	Indicates to use the event number from the last successful event task (e.g., the last event that was added or updated).
doc_source	Specifies the outside documentation source. The DD statement for the source dataset must be specified in the job's JCL or the Zeke procedure. **Note:** The Panvalet and Librarian options must be specified in the Zeke generation options to enable you to copy documentation from those sources. These are the valid values: BIMLibrary — Specifies the Bim-Edit library from which you want to copy documentation, followed by the parameter BIMMEMBER. CONDORMember — Specifies the Condor library member from which you want to copy documentation and can be followed by the parameters CONDORVER and/or CONDORPASS. DDNAME — Required for a partitioned or sequential dataset. Specifies the ddname that contains the member. DISK — Specifies the disk dataset from which you want to copy documentation, and can be followed by the parameters DDNAME, UNIT, LRECL, and BLOCK. INSTREAM — Indicates that the documentation follows as SYSIN data. All SYSIN documentation data must begin with the *doc*START statement and end with the *doc*STOP statement:

Parameter	Description	
	LIBRARIAN	Specifies the CA Librarian member name from which you want to copy documentation.
	PANVALET	Specifies the CA Panvalet member name from which you want to copy documentation.
	PDS	Specifies the partitioned dataset member name from which you want to copy documentation. The parameter DDNAME is required. For example: `DOC REPLACE 60 PDS DOC2 DDNAME DOCLIB`
	SEQUENTIAL	Required if you are copying sequential data. Specifies the DDNAME that contains the member.
	TAPE	Specifies the tape dataset from which you want to copy documentation, and can be followed by the parameters DDNAME, UNIT, LRECL, and BLOCK.
TEXT	Default. Specifies text documentation as the type of documentation to be added, replaced, or deleted. For example, this command deletes only text documentation data from event 50: `DOC DELETE 50`	
*doc*START	Indicates the beginning of SYSIN documentation where *doc* is one of these documentation types:	
	TEXT	Text documentation.
	DSN	Dataset documentation.
	NOTE	Note documentation.
	SCRATCH	Scratch pad documentation.
	Each *doc*START parameter must have a corresponding *doc*STOP parameter.	
statements	Specifies the documentation to be added, replaced, or deleted.	
	Note: For a DOC REPLACE command, if no statements are specified between the *doc*START and *doc*STOP parameters, then all existing documentation of that type is deleted.	

Parameter	Description
	These are the maximum line lengths and amount of data for each documentation type: **Type** **Columns** **Statements** TEXT 80 unlimited NOTE 60 ten statements SCRATCH 60 ten statements DSN 52 unlimited
*doc*STOP	Indicates the end of SYSIN documentation, where *doc* is one of these documentation types: TEXT Text documentation. DSN Dataset documentation. NOTE Note documentation. SCRATCH Scratch pad documentation. Each *doc*STOP parameter must have a corresponding *doc*START parameter.
ALL	Ignored if used with ADD or REPLACE. Specifies that you want to delete all documentation types. For example, this command deletes all documentation data (i.e., text, note, scratch pad, and dataset) from event 25: `DOC DELETE 25 ALL`

Parameter	Description

DSN	Specifies dataset documentation as the type of documentation to be added, replaced, or deleted. This parameter corresponds to the Tape field in the Documentation section of the EMR. Use this format:
	`I/O T/D VER DSN`
	where the DSN statements begin in column 1 and are formatted like this:
	• I/O is a single character in column 1.
	• T/D is a single character in column 3.
	• VER is a 3-character numeric, starting in column 5.
	• DSN is up to 44 characters, starting in column 9.
	For example:
	`I T 001 TEST.TAPEFILE.INPUT`
	`O D 000 TEST.DASDFILE.OUTPUT`

NOTE	Specifies note documentation as the type of documentation to be added, replaced, or deleted.
SCRATCH	Specifies scratch pad documentation as the type of documentation to be added, replaced, or deleted.

Sample JCL

This sample JCL illustrates the use of several parameters of the DOC function:

```
//STEP1    EXEC ZEKEUTL
//PDSDOC   DD DSN=PDS.DOC.TO.ZEKE,DISP=SHR
//SEQDOC   DD DSN=SEQ.DOC.TO.ZEKE,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
DOC ADD 34 LIBRARIAN DOCMEM1       Adds DOC TEXT to event 34 in the Zeke database
                                   if DOCMEM1 is found in the Librarian database
                                   specified in the Zeke started task procedure.

DOC ADD 35 PANVALET DOCMEM2        Adds DOC TEXT to Event 35 if DOCMEM2 is found
                                   in the Panvalet database specified in the
                                   Zeke started task procedure.

DOC ADD 36 PDS DOCMEM3 DDNAME PDSDOC
                                   Adds DOC TEXT from DOCMEM3 to event 36 if
                                   DOCMEM3 is in the PDS library specified by
                                   the DD=PDSDOC in the "add job" JCL.

DOC ADD 37 DDNAME SEQDOC SEQUENTIAL
                                   Adds DOC TEXT from the sequential dataset to
                                   event 37 specified by the parm DDNAME,
                                   provided the DDNAME is in the "add job" JCL.
```

```
DOC REPLACE 38                          Replaces scratch pad, text, and
                                        DSN documentation in event 38.
SCRATCHSTART
THIS IS SCRATCH AREA
ENTERED VIA BATCH
REPLACE COMMAND
SCRATCHSTOP
TEXTSTART
THIS IS TEXT DOC
ENTERED VIA BATCH
REPLACE COMMAND
TEXTSTOP
DSNSTART
I T 001 ZEKE.FILE.NAME
I D 002 ZEKE.FILE.NAME2
I T 005 ZEKE.FILE.REPLACED
DSNSTOP
/*
```

You can update all of an EMR's documentation sections in one step. For example, this sample SYSIN deletes the scratch pad documentation for event 1.

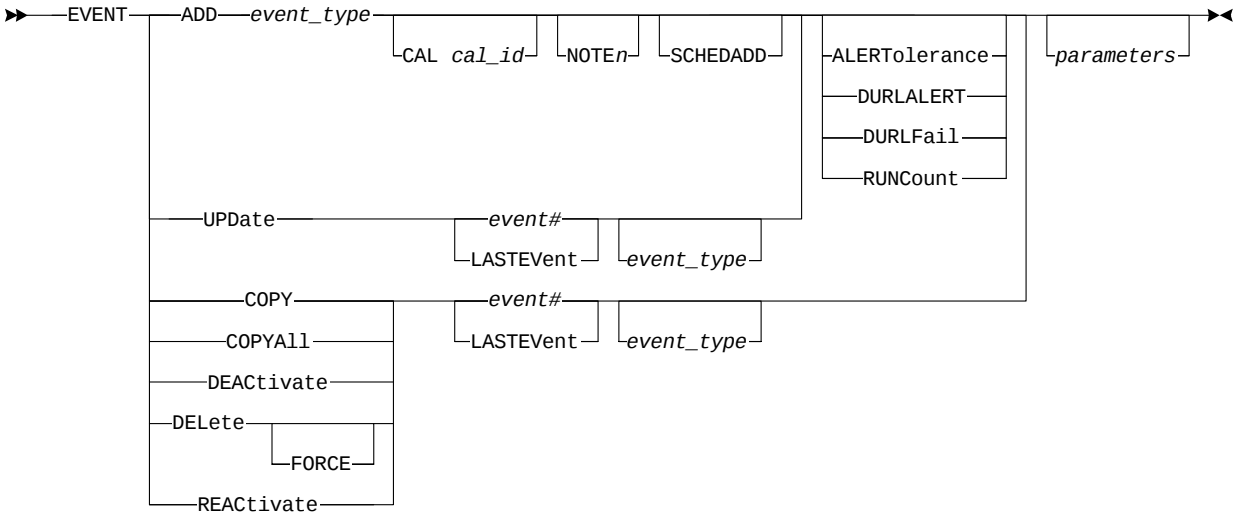
```
//STEP1    EXEC ZEKEUTL
//PDSDOC   DD DSN=PDS.DOC.TO.ZEKE,DISP=SHR
//SEQDOC   DD DSN=SEQ.DOC.TO.ZEKE,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
DOC REPLACE 1
TEXTSTART
THIS IS TEXT LINE 1 FROM A BATCH JOB
THIS IS TEXT LINE 2 FROM A BATCH JOB
THIS IS TEXT LINE 3 FROM A BATCH JOB
THIS IS TEXT LINE 4 FROM A BATCH JOB
TEXTSTOP
DSNSTART
I T 005 TAPE.INPUT.FILE.FIVE
O D 001 DISK.OUTPUT.ONE
I D 999 DISK.INPUT.999
DSNSTOP
SCRATCHSTART
SCRATCHSTOP
NOTESTART
THIS IS NOTE PAD LINE 1
THIS IS NOTE PAD LINE 2
THIS IS NOTE PAD LINE 3
NOTESTOP
/*
```

EVENT—Add/Maintain Event Master Records

The EVENT function enables you to add or update an event master record (EMR). This function provides most of the same options that are available when you define an EMR using the Zeke online facility.

See the *ASG-Zeke Scheduling for z/OS User's Guide* for more information on defining events, and adding events to the schedule as they are created.

Syntax



Parameters

This table describes the parameters that you can use with the EVENT function. The parameters are grouped by event type and purpose:

Parameter	Description
Required Parameters	
You must include one of these required parameters with the EVENT function to indicate the action to perform:	
ADD	Adds a new EMR to the Zeke database. For example: `EVENT ADD JOB XYZ...` This optional parameter is valid with the ADD function:

Parameter	Description	

	SCHEDADD	Adds the new event to the schedule queue as soon as it is defined to the Zeke database. For example:
		`EVENT ADD JOB TESTJOB1 PDS PRODLIB2 MEM TESTJCL2 OCC (MONDAY) SCHEDADD`
		For events with multiple versions, this parameter loads the number of versions specified in the Verload field on the EMR. If Verload is set to `0`, then SCHEDADD will schedule a version zero SQR only.
		SCHEDADD accepts operands. Almost all ZADD parameters can be coded as operands to SCHEDADD (see Notes, below). For example:
		`SCHEDADD 'CURRPLUS 0130'`
		This command updates the scheduled time on the added event by adding one hour and 30 minutes to the event's current Sched time. If the Sched time is zero, the specified time is added to the current system time and saved in the Sched time field.
		Notes:
		• Event number cannot be used with SCHEDADD as it takes the current event added by EVENT ADD function.
		• Version parameter cannot be coded with SCHEDADD when VERLOAD field on the EMR is greater than 1. See "VERLOAD" on page 44 for more information.

`COPY`	Copies the base definition of the specified EMR to create a new EMR. For example:
	`EVENT COPY 417`
	`EVENT COPY LASTEVENT`	

`COPYAll`	Copies the base definition of the specified EMR and all other associated event definition records (e.g., documentation, JCL, and resource information) to create a new EMR.	

`DEACtivate`	Flags an event as inactive, but does not remove it from the Zeke database. For example:
	`EVENT DEAC 417`
	`EVENT DEAC LASTEVENT`	

`DELete`	Deletes an EMR from the Zeke database. The event remains in the schedules of all active systems. For example:
	`EVENT DELETE 417`
	`EVENT DELETE LASTEVENT`
	This optional parameter is valid with the DELETE function:	

Parameter	Description
FORCE	Optional. You use this parameter to remove an event from the Zeke database forcibly in these cases: • If an event is corrupted and cannot be deleted through Zeke online system or the EVENT DELETE command. • If the event is a task event (i.e., that defines a remote Zena task). **Caution!** Use of this parameter orphans any database block used by the event. (The BACKUP function issues message Z2D66W for any block that is orphaned.) The deleted event number is marked as *reserved* and is not reused when a new event is added. A logical restore removes the reserve. You can recover an event that was deleted forcibly so it is re-added to the database with the same event number. The RECOVER command restores the event from a backup and removes the reserve (see "RECOVER—Restore Events from Backup" on page 83 for details). If an event that was deleted forcibly had any schedule records, those records are not removed from the schedule tables automatically and will continue to be displayed in Schedule View, or as ZDISPLAY command output. You can use the ZDELETE SQT command (which must be issued on each system sharing the Zeke database) to remove the event from the schedule tables. See "ZDELETE Command" on page 300 for details. You also can re-cycle Zeke using ZKILL COLD or TRACK (which rebuilds the schedule tables) to remove the event from the schedule tables.

REACtivate	Re-activates an inactive event, so that Zeke can include it in the schedule. For example:
	`EVENT REAC 311`
	`EVENT REAC LASTEVENT`

UPDate	Updates the definition of an existing EMR. For example:
	`EVENT UPDATE 45 PRIORITY 4`
	`EVENT UPDATE LASTEVENT PRIORITY 4`
	This optional parameter is valid with the UPDATE function:

Event Definition Parameters

These are the valid parameters that you can use with the EVENT function to define or update events:

event_type	Required with the EVENT ADD function to add a new event to the Zeke database. Specifies the event type. Once you add an event, you cannot change the event type. Use this parameter with the EVENT UPDATE function to change the attributes that specific to an event type (e.g., jobname, Zeke command, message text, etc.).

Parameter	Description
	These are the valid event types:

JOB	Indicates a job event (i.e., a JCL jobstream). You must specify the jobname (as entered on the job card). This is the name that is used by Zeke to track the event during execution and that is displayed on screens and in messages. For example:
	`EVENT UPDATE 34 JOB OPS7TX01`
	Several EVENT function parameters (e.g., resource requirements) are unique to job events (see "Job Event Parameters" on page 45 for a list).
	For a job event, one these Zeke JCL source parameters is required:
	`BIMLIBRARY`
	`BIMMEMBER`
	`CONDORMEMBER`
	`CONDORPASS`
	`CONDORVERSION`
	`FNAME`
	`FTYPE`
	`ICCFMEMBER`
	`ICCFLIBNO`
	`JCLMAN`
	`JCLMOD`
	`JCLPASS`
	`JESQJCL`
	`LIBRARIAN`
	`PANVALET`
	`PDS/MEM`

Parameter	Description
	`z14`*`cname`* For JCL from a Zeke-unsupported source. `ZEKEJCL` **Note:** If no JCL source parameters are specified, then no JCL source is available when job is ready for dispatch and Zeke places the job event on hold.
`MSG`	Indicates a message event (i.e., a message issued to the system console). You must specify the message to issue to the system operator console when the event is dispatched. Enclose the message in character string delimiters. For example: `EVENT ADD MSG 'LINE 1 OF MESSAGE'`

`MSG`*`n`*	Specifies the message text for additional lines using `MSG2` through `MSG6`.
	
	To clear a message line, enter the appropriate message parameter and at least one space between delimiters. For example:
	
	`MSG1 '  '`
	`MSG2 'MORE TEXT HERE IF NEEDED'`
	`MSG3 'STILL MORE, UP TO 6 LINES'`
	
	`EV UPD 438 MSG4 'NEW MESSAGE LINE'`

`SCOM`*`n`*	Indicates any console command event (any command issued from an operating system console. A system command (SCOM) event consists of up to six lines of commands and replies. Generally, the first line of an SCOM event is a command that causes the system to issue a message to which the SCOM event replies. Each line of an SCOM event specifies a command type. You specify the lines of the command text using `SCOM1` through `SCOM6`. For example: `EVENT ADD SCOM1 "C,D LST,ALL"` These are the valid command type codes: `C` System command `Z` Zeke command

Parameter	Description		
		V	VM command
		R	System response
		P	VSE/POWER command
	VCOM	Indicates a VM CP command event. You can specify any command (where the privilege class of the command and the issuing machine coincide). Enclose the command in character string delimiters. For example: `EVENT ADD VCOM (CP DRAIN 00E)`	

	ZCOM	Indicates a Zeke command event. You can specify any Zeke operator command or combination of commands. Enclose the command in character string delimiters. For example:
		`EVENT ADD ZCOM 'ZRELEASE JOB ZBC'`
		`EVENT ADD ZCOM 'ZR JOB ABC ZR JOB XYZ ZD WAIT'`
		Multiple commands are executed in sequence.	

	REXx	Indicates a REXX event (i.e., an event used to execute REXX execs. Specify the member name of the exec to be dispatched. For example: `EVENT ADD REXX "TESTREXX" REXXPRI=3 REXXCLASS=A` Zeke calls the REXX program as a function. A RETURN statement with an explicit return code is required. • If the return code is 0, Zeke will consider the REXX event to be successful. • If the return code in the RETURN statement is nonzero, is unspecified, or if the program exits without a RETURN statement, then Zeke will consider the REXX event to have failed. Some parameters are unique to REXX events (e.g., `REXXCLass`, `REXXPRIority`, and `ARGuments`).	
	TASK	Indicates a task event to be submitted to a remote Zena agent. You must specify the type of task. When you define a task event, you must specify the event name, remote system name, and the login profile (if required). Additional parameters are available based on the task type. These are the valid values:	
		SFTP	Secure File Transfer Protocol (sFTP). This task type requires an associated login record.

Parameter	Description		
		`EMAIL`	Submit an email. This task type requires an associated login record.
	`WORKcenter`	Indicates a work center event (i.e., a manual task that must be executed, but is not controlled by Zeke).	
		You specify the lines of the comment text using this parameter:	

		`LINE`*n*	Specifies the comment lines (up to 60 characters per line, where *n* is the line number ranging from `1` through `6`) to describe the work center activity. Enclose each comment line within character string delimiters. For example:
			`EVENT ADD WORK`
			`  LINE1 'THIS IS A HOLIDAY'`
			`  LINE2 'WORK CENTER EVENT'`
			When updating comment lines, you specify only the lines to be changed.
			Blank lines are useful for improving the readability of comments. To blank out a comment line, enter the appropriate LINE parameter and one space between delimiters. For example:
			`LINE1 'FRIDAY EMR UPDATE'`
			`LINE2 'FORMS DUE FROM ACCTNG'`
			`LINE3 ' '`
			`LINE4 'NOTIFY BY 4:30 PM'`
			`LINE5 'IF NOT RECEIVED'`

event#	Required for all functions (except EVENT ADD). Specifies the record on which to take the specified action. You also can use the LASTEVENT parameter in place of the actual event number. **Note:** The EVENT ADD function determines the event number and displays it on the SYSLST/SYSLOG. Event numbers are assigned in sequential order. Deleted event numbers are reassigned, but deactivated event numbers are not reassigned.		
`LASTEVent`	When used instead of the actual event number, this parameter indicates to use the same event number from the last successful event operation (e.g., the last event that was added or updated).		
`APPLIcat`	Specifies the user-defined code (up to eight characters long) that identifies the application with which the event is associated. This ID is used in reporting, work center control, and Zeke operator commands.		

Parameter	Description

DESCription
DESC2
	Specifies a one to two-line event description (up to 60 characters per line) to be used on summary screens and printed on reports. For a task event, the first description line is used as the Zena process description. Enclose the description within character string delimiters. For example:
	`DESC 'PAYROLL UPDATE FOR XYZ COMPANY'`
	`DESC2=/NOTE THE USE OF A VALID CHARACTER STRING DELIMITER/`

ENAme	Required with the ADD function for task events; optional for all other event types. Specifies the name of the event. This is the name that will display on screens and reports to help you identify the event, and also is how events are referenced in end-of-event (EOE) WHEN conditions and Zeke operator commands. For a task event, this defines the Zena process name.
GROupid	Specifies the user-defined code (up to three characters long) to identify the group with which the event is associated. This value is used in reporting, work center control, and Zeke operator commands.

NOTE*n*	Valid with the ADD function only. Specifies the notes (up to 60 characters per line, where *n* is the line number ranging from `1` through `10`) for the notepad area of the event. Enclose each note within character string delimiters.
	You can use blank lines to improve the readability of the notepad. To do so, enter the appropriate NOTE parameter and one space between delimiters. For example:
	`NOTE1 'THIS EVENT IS THE MAIN KICKOFF EVENT IN THE PAYROLL SERIES.'`
	`NOTE2 'NOTIFICATION OF WHEN TO RUN THIS EVENT WILL COME BY FAX FROM'`
	`NOTE3 ' '`
	`NOTE4=.NOTICE THAT YOU CAN USE OTHER DELIMITERS.`

TEMPlate	Specifies the code to indicate whether the event is used as a model to create other events of the same type. **Note:** A template cannot be scheduled like a normal event (even if it is activated and has a defined OCCURS clause). These are the valid values: `N` The event is not a template. `Y` The event is a template, which can be used as a model to create other events of the same type.

Parameter	Description
SYStemid	Specifies the name (up to eight characters long) to identify the system or pool that owns the event. Or, you can specify *ANY (which Zeke interprets as a *pool* of all Zekes in the Zekeplex and enables the event to be dispatched on any of the Zeke systems). An event is associated with only one system or pool. The default value is A. For example: SYS PROD SYS A
USERid	Specifies the user-defined code (up to eight characters long) to identify the user associated with the event. This value is used in reporting, work center control, and online security. For example: EVENT ADD COMMENT USERID DATACTL1... EVENT ADD JOB ABC USERID OPERATOR... To remove a user ID on an EMR, you can use this parameter and blank out the user ID. For example: EVENT UPDATE 123 USERID ' '

Event Scheduling and Dispatching Parameters

These are the valid parameters that you can use with the EVENT function to specify event scheduling and dispatching information:

AVGDURation	Specifies the average duration for the event in *hhhh:mm:ss* format.
CALid	Specifies the calendar ID to use when scheduling the event. The default value (if you are adding an event) is A. The calendar must exist before the SCHEDULE function is run. Use the CALENDAR function (see "CALENDAR—Add/Maintain System Calendars" on page 14), or the Zeke online facility, to define the calendar. For example: CAL PROD
CONTrol	Specifies the code to indicate whether this job is tracked as a Zeke-controlled event (which are tracked throughout the entire execution). These are the valid values: YES — Default. Zeke recognizes this event as an Zeke-controlled event. NO — Zeke does not recognize this event as an Zeke-controlled event and automatically updates the event to a successful status when the event dispatched. NX — Zeke recognizes the event as a non-executable Zeke-controlled event. Non-executable events are scheduled like any other event and are useful as predecessors to other events. A non-executable event is never submitted to JES for JCL execution. Zeke automatically updates the event to a successful status when the event dispatched and any dependent events are triggered.

Parameter	Description
	Note: If you want to remove an event from an intricate event flow, you simply can mark the event as non-executable. You can do this as an alternative to updating the event flow, which typically requires deleting the event and then modifying the WHEN clauses for all of the deleted event's successors. See the *ASG-Zeke Scheduling for z/OSVSE User's Guide* for more information on non-executable events.
`DPRiority`	Specifies the Zeke dispatch priority number ranging from `0` (highest) through `99` (lowest) to distinguish priority event from normally scheduled events. The default value is `50`.
`DRL`	Specifies a Zeke disaster recovery level for the event, if desired.

`EARLYtime`	Specifies the earliest time (in *`hh:mm`* format) that Zeke can dispatch this event. The valid values range from `00:00` through `47:59`. If the time is greater than `24:00`, then Zeke considers the event to require dispatching the next day. For example:
	`EARLYTIME 01:00`
	This event is eligible for dispatch at 1:00 A.M.
	`EARLY 16:00`
	This event is eligible for dispatch at 4:00 P.M.
	Note:
	To remove the early time, you can specify EARLY without an operand.
	Note:
	It is possible for Zeke to dispatch an event at its early time; however, Zeke still dispatches the event in sequence according to its schedule time.

`EXPIre`	Specifies the date (*`MMDDYYYY`* or *`DDMMYYYY`*) that the event expires. When the SCHEDULE function runs (with the ACTIVATE parameter) on or after this date, the event is deleted automatically. For example:
	`EXPIRE 10312020`
	This deletes event on or after October 31, 2020.
	To remove the expiration date, specify `0` (zero). For example:
	`EXPIRE 0`
	Note:
	If a schedule forecast is run for a date that is after the expiration date, a message informs you of the expired event. The event is not deleted.

Parameter	Description
FREQCalc	Specifies the code to indicate how to calculate the next dispatch time. These are the valid values: C (Clock/system time) The next run is based on the completion time of the previous run. S (Schedule time) Default. The next run is based on schedule time regardless of when the job actually runs.
LATEEND	Specifies the time by which the event must finish (in *hh:mm* format). The valid values range from 00:00 through 47:59. If the 'late end' time is reached and the event has not completed yet, then Zeke issues message Z0302I to the console and flags the event with a late status. An event that is projected to finish late is not assigned a late status until the 'late end' time is reached. An event that is projected to finish late is not prevented from being dispatched until its 'must end' time is violated. If an event has both a 'late start' and a 'late end' time, the condition that occurs first will cause the event to be flagged as late and message Z0302I to be issued. If the 'late end' time is greater than 24:00, then Zeke considers the event to require dispatching the next day. For example: LATEEND 20:00 Notifies the operator if the event is not projected to finish by 8:00 P.M. LATEEND 28:00 Notifies the operator if the event is not projected to finish by 4:00 A.M. Late events can have a higher dispatch priority based on the PriLate generation option (see page 568). To remove the 'late end' time, specify LATEEND without an operand.
LATEstart	Specifies the time (in *hh:mm* format) by which the event must be dispatched. The valid values range from 00:00 through 47:59. If the 'late start' time is reached but the event has not dispatched yet, then message Z0302I is issued to the console. An event projected to start late is not assigned late status until the 'late start' time is reached. An event projected to start late is not prevented from being dispatched unless its 'not after' time is violated. If an event has both a 'late start' and 'late end' time, the condition that occurs first will cause the event to be flagged as late and message Z0302I to be issued. If the event is not projected to be dispatched by the 'late start' time (based on its predecessors), an early warning alert is issued to OpsCentral.

Parameter	Description

	If the 'late start' time is greater than `24:00`, Zeke considers the event to require dispatching the next day. For example:
	`LATESTART 20:00`
	Notifies the operator if the event is not dispatched by 8:00 P.M.
	`LATESTART 28:00`
	Notifies the operator if the event is not dispatched by 4:00 A.M.
	Late events can have a higher dispatch priority based on the PriLate generation option (see page 568).
	To remove the late time, specify `LATESTART` without an operand.

`MILESTone`	Specifies the code to indicate whether the event is a milestone event. A *milestone event* is a significant event in the event flow (which includes predecessor/successor relationships) that must be processed on time to avoid a significant delay in the completion of the entire event flow. Events flagged as milestones are not processed any differently from other events—the milestone flag simply makes these events easier to identify in the event flow. These are the valid values: `NO` Default. Do not flag the event as a milestone event. `YES` Flag the event as a milestone event. See the *ASG-Zeke Scheduling for z/OS User's Guide* for more information on milestones.
`MULThit`	Specifies the code to indicate whether to enable an event to be scheduled multiple times due to a nonworking day. `N` Do not enable an event to be scheduled multiple times due to a nonworking day. The number of times defaults to the value specified in the MultHit generation option (see page 557). `Y` Default. Allow an event to be scheduled multiple times due to a nonworking day.
`MUSTend`	Specifies the latest time that the event can complete processing (in *`hh:mm`* format). The valid values range from `00:00` through `47:59`. If the 'must end' time is less than the calculated time, the event is put on hold and a message is issued to the operator console.
`NOTAFter`	Specifies the latest time (in *`hh:mm`* format) that Zeke can dispatch the event. The valid values range from `00:00` through `47:59`. If the 'not after' time is less than the system time, the event is put on hold and a message is issued to the operator console.

Parameter	Description	
NWDAY	Specifies the code to indicate how to schedule this event on nonworking days. These are the valid values:	
	A	Default. Schedule the event after the nonworking day.
	B	Schedule the event before the nonworking day.
	N	Do not schedule the event.
	O	Schedule the event on the nonworking day.
OCCURS	Specifies an OCCURS clause (up to 500 characters long) indicating the conditions for when the event should be added to the schedule. `EVENT ADD JOB OCCURS (TUESDAY)` See the *ASG-Zeke Scheduling for z/OS User's Guide* for instructions on how to define OCCURS clauses and for a listing of OCCURS clause keywords.	
OPERok	Specifies whether Zeke is to wait for an operator approval before dispatching the event. These are the valid values:	
	NO	Zeke automatically dispatches the event.
	YES	Zeke dispatches the event only when dispatching has been approved by Zeke operator command. When the event is moved from the schedule queue to the dispatch queue, a message is issued to the operator console requesting the approval. The wait reason AWAITING OPER OK is displayed on ZDISPLAY screens.
PERManent	Specifies the code to indicate whether the event is to remain in the schedule permanently (i.e., indefinitely). These are the valid values:	
	N	Default. The event is not considered a permanent event and is processed by each schedule run according to its OCCURS clause.
	Y	The event is marked as permanent and its OCCURS clause is set to `REQUEST`. The event must be added to the schedule using the ZADD command. When the event is added, it remains in the schedule even across schedule runs. You can delete the event only by using the ZDELETE command. For permanent events, do not set a Times value (see page 43); permanent events can run an unlimited number of times. If a permanent event is later changed to be non-permanent, Zeke automatically sets the Times value to `1`. Also, the Trig value (see page 43) for permanent events always is set to `ALL`.

Parameter	Description
	See the *ASG-Zeke Scheduling for z/OS User's Guide* for more information on permanent events.
RETAin	Specifies the code to indicate whether Zeke is to retain the event if it is not dispatched on the scheduled date. These are the valid values: `NO` Discard the event if it is not dispatched. `YES` Default. Retain the event for the next run.

SCHEdtime	Specifies the normal schedule time (in *hh:mm* format) for this event. The valid values range from `00:00` (default) through `47:59`. If the time is greater than `24:00`, Zeke considers the event to require dispatching the next day. For example:
	`SCHED 12:00`
	This event is scheduled for 12:00 P.M. (noon).
	`SCHED 24:00`
	This event is scheduled for 12:00 A.M. (midnight).
	`SCHED 30:00`
	This event is scheduled for 6:00 A.M. the next day.
	Note:
	It is possible for Zeke to dispatch an event at its early time; however, Zeke still dispatches the event in sequence according to its schedule time.

SCHENV	Specify the name (up to 16 characters long) of the WLM scheduling environment. (Zeke does not validate this name; an invalid name will cause JES to fail the job.) If you specify a scheduling environment, the event is scheduled and waits in the schedule queue until the scheduling environment is active (i.e., until the resource states defined in the scheduling environment are matched). When the scheduling environment becomes active, the event is dispatched. **Note:** Optionally, for a job event, this value can be inserted in the JCL before the job is submitted to JES. See the *ASG-Zeke Scheduling for z/OS User's Guide* for more information on overriding the JOB card. To remove the scheduling environment, specify SCHENV without a value.

Parameter	Description
SET	For work center events, this parameter specifies the variables and associated values (up to 500 characters long) to complete the event. **Note:** See the *ASG-Zeke Scheduling for z/OS User's Guide* for more information on using variables for work centers, and how to define a SET clause. For other event types, this parameter performs the same function as the WHEN parameter. You specify any prerequisites (up to 1,360 characters) that must occur before Zeke dispatches the event.
TIMES	Specifies the number of times (from 1 through 9999) to dispatch this event per schedule run. The default value is 1. For example: TIMES=3 This event occurs three times. If this value is greater than 1, the event is considered a *recurring* event. See the *ASG-Zeke Scheduling for z/OS User's Guide* for more information on recurring events. **Note:** For permanent events, do not set a Times value; permanent events can run an unlimited number of times. If a permanent event is later changed to be non-permanent, Zeke automatically sets the Times value to 1.
TARGet	Valid for job and task events only. Required for task events. For a job event, the TARGET value specifies the name of the remote system (i.e., Zeke or Zeke Agent) where the job event will be submitted (as opposed to the system ID of the dispatching system). The target name for a remote Zeke system is defined through the Netregid generation option (page 561) for that system. The target name for a remote Zeke agent is defined as the Netregid in the agent's configuration. For a task event, the TARGET value specifies the name of the remote Zena agent where a task event will be submitted for execution. The target name for a Zena agent is defined through the agent properties in Zena (see the *ASG-Zena User's Guide* for details). If this parameter is omitted, the default value is *LOCAL.
TRIG	Valid for recurring events only (i.e., an event with multiple occurrences within a schedule run). Specifies a code to indicate when the recurring event can satisfy WHEN conditions (i.e., serve as a trigger) for other events. These are the valid values: A — Default. The recurring event can trigger other events each time it runs.

Parameter	Description
	Note: Permanent events (i.e., recurring events which can occur an unlimited number of times) always trigger on *all* occurrences. See the *ASG-Zeke Scheduling for z/OS User's Guide* for more information on permanent events. `F` The recurring event can trigger other events only the first time it runs. `L` The recurring event can trigger other events only the last time it runs. For example, a recurring event is scheduled to run every hour starting at 8:00 A.M. and ending at 12:00 P.M. • If Trig is set to `F`, the event can satisfy WHEN conditions only on the 8:00 A.M. run, and all subsequent trigger calls for this event are ignored (until the event is rebuilt or refreshed). • If Trig is set to `L`, the event can satisfy WHEN conditions only on the 12:00 P.M. run, and all other trigger calls for this event are ignored (until the event is rebuilt or refreshed). • If Trig is set to `A`, the event can satisfy WHEN conditions on each run from 8:00 A.M. to 12:00 P.M. **Note:** For a non-recurring event (i.e., the TIMES value is `1`), the event will trigger regardless of the setting of the TRIG parameter.
`VERLOAD`	Specifies the number of versions of this event to load during the schedule build. The default value is `0`. For example: `EVENT UPDATE VERLOAD 00005` • If VERLOAD is set to `0`, only one version of the event (version zero) can be in the schedule at a time. For permanent events, a VERLOAD value is not allowed (only one version of a permanent event can be in the schedule). • If VERLOAD is set to `1`, only one version is created by the schedule build, but any number of versions (up to 32,767) can be added to the schedule after schedule load using the ZADD command. **Note:** ASG recommends running no more than 1,000 versions of a single event.

Parameter	Description

WHEN	For event types other than work centers, this parameter specifies any prerequisites (up to 1,360 characters) that must occur before Zeke dispatches the event. For example:
	`EVENT ADD JOB WHEN (EOJ JOBA)`
	This adds a new job event with a default WHEN condition of `EOJ JOBA`.
	`EVENT UPDATE 9 WHEN (EOJ JOBB)`
	This updates the default WHEN condition for event 9.
	See the *ASG-Zeke Scheduling for z/OS User's Guide* for how to define WHEN conditions, and for a listing of WHEN condition keywords.
	For work centers, this parameter performs the same function as the SET parameter. You specify the variables and associated values (up to 500 characters) to complete the event.
	Note:
	By default, this parameter maintains the WHEN condition for the default version (i.e., version 0) of the event. Include the WHENVER parameter to operate on a WHEN condition for a different version.

WHENVER	Valid with the WHEN parameter. Specifies the version of the WHEN condition to be updated by the WHEN parameter. If omitted, the default version is `0`; the WHEN parameter updates the WHEN condition for the default version (i.e., version 0). For example: `EVENT UPDATE 9 WHENVER 2 WHEN (EOJ JOBVER2)` This example updates the WHEN condition for version 2 of event 9.

Job Event Parameters

These are the valid parameters that you can use with the EVENT function for job events only:

ALERTOLerance	Valid with the EVENT ADD or EVENT UPDATE function for job events only. Specify a value for Zeke to use to calculate the acceptable range of duration times (indicated in the Normal Range field on the EMR) for the event. The valid values range from `0` through `100`. If alerts are enabled for this event, executions that run shorter or longer than the normal range value will generate an alert. See "Alert Tolerance" on page 427 for more information.
BIMLibrary	Valid for job events only. Specifies the Bim-Edit library that contains the appropriate JCL.
BIMMember	Valid for job events only. Specifies the Bim-Edit member name.

Parameter	Description

`CLAss`	Valid for job events only. Specifies up to six classes for the event. The valid values range from `A` through `Z`, and `0` through `9`.
	For example:
	`EVENT UPD 62 CLA A`
	`EVENT UPD 62 CLA (A,B,C)`
	`EVENT ADD JOB TEST CLASS (G,H,I,T,J,K)`
	Note:
	To remove the class, specify CLASS without an operand.
	See the section, "Initiator Processing" in Chapter 6, "Resources" in the *ASG-Zeke Scheduling for z/OS User's Guide* for more information.

`CONVert`	Valid only with the UPDATE function for job events only. Use this parameter with the PLATFORM parameter to specify a new platform type (see page 49 for valid values) for the event (when you want to convert the event from one platform to another).
`CONDORMember`	Valid for job events only. Specifies the Condor library member that contains the appropriate JCL.
`CONDORVersion`	Valid for job events only. Specifies the Condor version number. If omitted, the default value is `001`.
`CONDORPass`	Valid for job events only. Specifies the Condor password. For example: `EVENT ADD JOB DOSNAME3 CONDORM MEMBNM2 CONDORP PASS`
`DURALERTS`	Valid with the ADD or UPDATE function for job events only. Use this option to override the DurAlert generation option for a particular event. The DurAlert generation option (see page 543) indicates whether to issue a console message and OpsCentral alerts if a job runs longer or shorter than the acceptable range of duration times. These are the valid values: `DEF` — Default. Use the DurAlert generation option value. `NO` — Disable duration alerts for this event. `YES` — Enable duration alerts for this event.
`DURFail`	Valid with the ADD or UPDATE function for job events only. Use this option to override the DurFail generation option for a particular event. The DurFail generation option (see page 544) indicates whether to fail jobs that run longer or shorter than the acceptable range of duration times (which prevents them from triggering successor jobs). These are the valid values: `DEF` — Default. Use the DurFail generation option value. `NO` — Do not fail this job if it runs longer or shorter than the acceptable range of duration times.

Parameter	Description
	YES — Fail this job if it runs longer or shorter than the acceptable range of duration times. **Note:** This parameter does not cause a job to be canceled. When a job ends, if its duration fell outside the normal range, Zeke marks the event as failed and issues message Z8T02I. If a job abends or fails due to a condition code record, Zeke does not mark the job as failed due to its duration. Zeke fails a job due to its duration only if it otherwise would have been marked as successful.
FName	Valid for job events only. Specifies the CMS filename that contains the appropriate JCL or exec that creates the JCL. For example: `EVENT ADD JOB JOBNAM8 FNAME USRNAM01`

FREQuency	Valid for recurring job events only. Specifies the time interval to wait before Zeke dispatches the event again, in (*hh*`:`*mm*) format. Zeke adds the current schedule time or the system time with this value to determine the next schedule time. The valid values range from `00:00` through `47:59`. For example:
	`FREQUENCY=01:00`
	Zeke dispatches the event every hour.
	`FREQ 00:30`
	Zeke dispatches the event every 30 minutes.

FTYPE	Valid for job events only. Specifies the CMS file type. If you specify `EXEC`, then Zeke assumes that the named value is an exec that creates the JCL. For example:
	`EVENT ADD JOBNAME9 FNAME USRNAM01 FTYPE OSJCL`
	`EVENT ADD JOBNAME9 FNAME USRNAM02 FTYPE EXEC`
	If omitted, this parameter defaults to the file type specified by the CmsFType generation option (see page 532).

JCLSTART	Valid for job events only. Indicates the beginning of the SYSIN JCL to be added for an event as part of the EVENT ADD process. All SYSIN JCL data must begin with the JCLSTART statement and end with the JCLSTOP statement. The SYSIN JCL must occur after the basic EMR data for the event. The SYSIN DD of the batch job to supply the EVENT ADD data must have this format: `//SYSIN DD DATA,DLM=@@` JCL SYSIN data is limited to 450 lines and 80 characters per line. Leading blanks are preserved for all statements not beginning with `//`, `/+`, `/-`, or `+##`. All JCL statements preceded by one of the characters listed above are automatically positioned in column 1.

Parameter	Description

	For example:
	`//SYSIN DD DATA,DLM=@@`
	`EVENT ADD JOB TESTJOB APP 'TST' GROUP 'TEST`
	`SCHED 10:00 ZEKEJCL=YES`
	`OCC (MONDAY)`
	`JCLSTART`
	`//TESTJOB JOB MSGLEVEL=(1,1)`
	`//STEP1 EXEC PGM=IEBCOPY`
	`//SYSPRINT DD SYSOUT=*`
	`//INDD DD DSN=DSA.TEST,DISP=SHR`
	`//OUTDD DD DSN=DSA.TEST2,DISP=SHR`
	`//SYSIN DD * COPY INDD=INDD,OUTDD=OUTDD`
	`/*`
	`JCLSTOP`
	`@@`
	
	Note:
	You can specify JCL sources other than ZEKEJCL when you add an event using the Zeke online facility (even if JCL SYSIN is present).

`JCLSTOP`	Valid for job events only. Indicates the end of the SYSIN JCL to be added for an event as part of the EVENT ADD process. See the JCLSTART parameter on page 47.
`JESQJCL`	Valid for job events only. Indicates that the event's JCL is contained in the JES job queue. The Target value automatically is set to `*LOCAL`. One of these values is required: `D` — A new SQR is created dynamically for a matching job that enters the JES queue (if an unassigned SQR does not already exist). If an SQR is created dynamically, then the job ID of the JES queue job is inserted in the SQR when it is created. At that point, the SQR is assigned to the JES queue job and is the only SQR that can dispatch the job. If JESQJCL is set to `D`, then the value for VERLOAD is automatically set to `1` so that Zeke creates multiple SQRs dynamically for the same schedule date. When Zeke creates an SQR dynamically, the value for the Retain field is set to `Y` and the Times field is set to `001`. If JESQJCL is set to `D`, the SCHEDULE function or the ZADD command can be used to add a job event. `Y` — The event's SQR must be created in the normal way (i.e., scheduled by the SCHEDULE function or added using the ZADD command).
`LIBRarian`	Valid for job events only. Specifies the CA Librarian library member name that contains the appropriate JCL.

Parameter	Description
MEM	Valid for job events only. Specify the PDS member name containing the event's JCL. For example: `EVENT ADD JOB TESTJOB2 PDS PRODJCL MEM TEST` **Note:** For ADD and UPDATE functions, if MEM is specified without the PDS parameter, the PDS field is changed to the default PDS name specified in the PdsDD generation option (see page 565).
PANvalet	Valid for job events only. Specifies the CA Panvalet library member name (up to ten characters long) that contains the appropriate JCL. For example: `EVENT ADD JOB JOBNAME3 PANVALET PANNAME1`

PDS	Valid for job events only. Specify the PDS ddname that is associated in the Zeke started task with the partitioned dataset containing this event's JCL. The member name is required with PDS. For example:
	`EVENT ADD JOB TESTJOB1 PDS PRODLIB2 MEM TESTJCL2`
	When using ZEKEJCL as the PDS name, it must be enclosed in quotation marks. For example:
	`EVENT ADD JOB TESTJOB1 PDS 'ZEKEJCL' MEM TESTJCL1`
	Caution! The ddname can be a Zeke variable (which would enable JCL to be retrieved from a different dataset) depending on the assigned value of the variable. This can cause performance degradation due to operating system overhead associated with dynamic dataset allocations.

PLATform	Valid for job events only. Specifies the platform on which the JCL is to be executed. These are the valid platforms: AIX DCOSX (Pyramid) HPUX MVS (includes z/OS) OS2 OS400 SUN TANDEM USYS UNIX (i.e., AIX, AT&T, HPUX, NCR, SCO, SunOS, Sun Solaris, etc.) VMS VSE WINDOWS (includes all supported versions)

Parameter	Description
	Note: Although the AIX, HPUX, and SUN platform codes listed above are supported, it is preferred that you use the UNIX platform code. If this parameter is omitted, the platform defaults to the value specified in the DefPltfm generation option (see page 537).
PRIority	Valid for job events only. Specifies the OPSYS priority code (from 1 through 99) that is valid for the platform on which this job is executed. For example: PRIORITY=5 If this parameter is omitted, the default priority code defaults to the value defined in the DefJPrty generation option (see page 536).
RUNCount	Valid with the ADD or UPDATE function for job events only. This option updates the 'Job ran __ Times' value on the EMR (which indicates the number of job runs included in the job's current duration statistics.) If you change this value to be less than the number indicated by the DurCount generation option (see page 544), then Zeke does not generate duration alerts or fail the job if it runs short or long. (Jobs are failed only if the DurFail generation option (see page 544) is set to Y or if the Fail option in the EMR is enabled.) **Note:** If the job has not run enough times to meet the DurCount setting, you can increase the 'Job ran __ Times' value to start generating duration alerts/failures sooner. Average duration (AvgDur) and normal range values that are calculated from only a few runs might not be truly representative of the job's normal duration, and so the alerts/failures generated might not be appropriate.
SECGROUP	Valid for job events only. Applicable if the RepJUser generation option (see page 573) is set to A or C. For job events on z/OS, enter the security group (up to eight characters long) to be inserted into the USER= keyword on the JOB statement when the job is submitted to z/OS.

Parameter	Description
TAPES	Valid for job events only. Specifies the number of tape drives required by this job (from 0 through 255). Zeke ensures that the specified number of tape drives is free before dispatching the job. If the job is ready to be dispatched and the required number of drives is not free, Zeke issues a message to the operator console that indicates the number of drives for which the job is waiting. For example, this parameter indicates that the job requires three tape drives: `TAPES=3` If the CalcTap generation option (see page 530) is set to Y, Zeke calculates this number based on the last run. You only need to specify a value if you want to override this value. **Note:** You can update this value in the SQR using a Zeke operator command, but this does not affect the EMR.
ZEKEJCL	Valid for job events only. Specifies whether the event's JCL resides in the Zeke database. These are the valid values: NO — The event's JCL does not reside in the Zeke database. YES — The event's JCL resides in the Zeke database. For example: `EVENT ADD JOB JOBNAMEX ZEKEJCL=YES` **Note:** You can add JCL to the Zeke database automatically through the EVENT ADD function by including the JCLSTART parameter (see page 47).
z14cname	Valid for job events only. Specifies the name (up to eight alphanumeric characters long) for the non-Zeke source (enabled by the ZEKE14C JCL user exit) followed by the member name. This user-defined source name is defined in the Zeke generation options X14Cnm1 (see page 604) and X14Cnm2 (see page 604) and is displayed on the Event Master Records Functions screen. In this example, USERNAME is the value that is specified for the X14Cnm1 generation option: `EVENT UPD 100 USERNAME TESTJCL`

Parameter	Description

Condition Code Processing Parameters

These are the valid parameters that you can use with the EVENT function to define condition code processing criteria for a job event:

`CCACTION` Specifies the code to indicate whether condition code processing is to be executed at EOJ. These are the valid values (which can used together with other CC parameter codes):

- `F` Flag the job as failed if the condition code criteria is met. For example:

  ```
  CCACTION F CCLOW 0004 CCOPER GT
  ```

  If the EOJ condition code is greater than `4`, then the job is considered to have failed.

- `C` Cancel the job if condition code criteria is met.

- `O` Continue processing if condition code criteria is met.

- `S` Valid for only for remote jobs sent to Zena. Request that Zeke receive an intermediate status (i.e., an EOS trigger) for the matching step.

Note:

If both a CCACTION value and a condition code list (indicated by CCSTART and CCSTOP) are specified, the CCACTION value is validated and processed, but is replaced by the condition code list.

This example shows multiple CC parameters used together:

```
CCACTION=A CCHIGH=12 CCLOW=4 CCOPER=RA CCSTEP=STEP1 CCPROC=USRPG1
```

`CCLOW` Specifies a condition code (up to four characters long) to serve as the low value for the condition code RANGE criteria, or a single value to be compared to a job's maximum EOJ condition code. (See the example provided for CCACTION on page 52.)

`CCHIGH` Specifies a condition code (up to four characters long) to serve as the high value for the condition code RANGE criteria. A CCHIGH value must be specified if the CCOPER parameter is set to `RA`. (See the example provided for CCACTION on page 52.)

`CCOPER` Specifies the operation to determine the comparison mode for the supplied CCLOW and CCHIGH values. These are the valid codes:

- `GT` Greater Than
- `EQ` EQual to
- `LE` Less than or Equal to
- `LT` Less Than

Parameter	Description		
	GE	Greater than or Equal to	
	NE	Not Equal to	
	RA	RAnge	
	(See the example provided for CCACTION on page 52.)		
CCPROC	Specifies the PROCSTEP name to be checked for a condition code. (See the example provided for CCACTION on page 52.)		

CCSTART	Indicates the beginning of the list of condition code definition statements that will replace the ones currently defined for the event. Use the CCSTOP parameter to indicate the end of the list.
	You can include only one block of statements for each execution of the EVENT function.
	If multiple condition codes are defined for an event, you cannot add or update an individual code; you must replace the existing list with a new list.
	For example:
	`EVENT ADD JOB NEWJOB ZEKEJCL YES`
	` CCSTART`
	`  STEPCC STEP STEP1  OPERATOR RA LOW 4 HIGH 9 ACT C`
	`  STEPCC STEP STEP2  OPERATOR GE VALUE 4 ACT F`
	`  STEPCC STEP STEP2  OPERATOR GT VALUE 8 ACT C`
	`  EOJCC OPER GE LOW 8 ACT F`
	` CCSTOP`		

CCSTEP	Specifies the to be checked for a condition code. (See the example provided for CCACTION on page 52.)		
CCSTOP	Indicates the end of the list of condition code definition statements that will replace the ones currently defined. Use with the CCSTART parameter (which indicates the beginning of the list).		
EOJCC	Specifies the end-of-job (EOJ) condition code (which is evaluated when the job ends from the maximum condition code from any of the job steps). Include this parameter at least once in a condition code list (defined by the CCSTART and CCSTOP parameters). These are the valid parameters:		
	ACTION	Action to be taken if the specified condition code rule is true. These are the valid actions:	
		F	Flag the job as failed during EOJ processing.
		O	Continue processing normally.
	LOW	Required if OPERATOR is set to RA. Low value for the RA operator.	
	HIGH	Required if OPERATOR is set to RA. High value for the RA operator.	

Parameter	Description		
	OPERATOR	Required. Comparison operator. These are the valid values:	
		EQ	Condition code is equal to the specified value.
		GT	Condition code is greater than the specified value.
		GE	Condition code is greater than or equal to the specified value.
		LE	Condition code is less than or equal to the specified value.
		LT	Condition code is less than the specified value.
		NE	Condition code is not equal to the specified value.
		RA	Condition code is within the specified range.
	VALUE	Required for all OPERATOR values except RA. The value to be compared against the step or EOJ condition code.	

For example:

```
EVENT ADD JOB NEWJOB ZEKEJCL YES
 CCSTART
  STEPCC STEP STEP1  OPERATOR RA LOW 4 HIGH 9 ACT C
  STEPCC STEP STEP2  OPERATOR GE VALUE 4 ACT F
  STEPCC STEP STEP2  OPERATOR GT VALUE 8 ACT C
  EOJCC OPER GE LOW 8 ACT F
 CCSTOP
```

Note:

If EOJCC is omitted, then no condition code checking is done at end-of-job.

STEPCC

Specifies the end-of-step condition code. When a job step ends, Zeke evaluates each step condition code rule in order. The first rule that matches (i.e., based on step name, procedure step, operator, and the job step's condition code) is used and the specified action is taken. These are the valid parameters:

ACTION	Action to be taken if the specified condition code rule is true. These are the valid actions:	
	C	Cancel the job at this step and flag it as failed at end-of-job (EOJ) processing.
	F	Flag the job as failed during EOJ processing.

Parameter	Description		
		O	Continue processing normally.
		S	Valid for only for remote jobs sent to Zena. Request that Zeke receive an intermediate status (i.e., an EOS trigger) for the matching step.
	LOW	Required if OPERATOR is set to RA. Low value for the RA operator.	
	HIGH	Required if OPERATOR is set to RA. High value for the RA operator.	
	OPERATOR	Required. Comparison operator. These are the valid values:	
		EQ	Condition code is equal to the specified value.
		GT	Condition code is greater than the specified value.
		GE	Condition code is greater than or equal to the specified value.
		LE	Condition code is less than or equal to the specified value.
		LT	Condition code is less than the specified value.
		NE	Condition code is not equal to the specified value.
		RA	Condition code is within the specified range.
	PROCSTEP	Specifies the procedure step name (up to eight characters long).	
	STEPNAME	Required. Specifies the step name (up to eight characters long).	
	VALUE	Required for all OPERATOR values except RA. The value to be compared against the step or EOJ condition code.	

For example:

```
EVENT ADD JOB NEWJOB ZEKEJCL YES
 CCSTART
  STEPCC STEP STEP1  OPERATOR RA LOW 4 HIGH 9 ACT C
  STEPCC STEP STEP2  OPERATOR GE VALUE 4 ACT F
  STEPCC STEP STEP2  OPERATOR GT VALUE 8 ACT C
  EOJCC OPER GE LOW 8 ACT F
 CCSTOP
```

Event Resource Parameters

These are the valid parameters that you can use with the EVENT function to specify event resource criteria:

Parameter	Description	
RESASSume	Specifies the code to indicate whether the event will obtain the resource from an abended event that has the RESKEEP parameter set to YES. These are the valid values:	
	NO	An event cannot steal the resource from an abended event that has RESKEEP set to YES. **Note:** If RESMODE (see page 57) is set to SR, then RESASSUME and RESKEEP must be set to NO).
	S	An event that abends can steal the resource from itself, but cannot steal it from another abended event.
	YES	An event can steal the resource from any abended event that has RESKEEP set to YES.
RESCount	Specifies the number of resources (from 1 through 999) that this event requires. The default value is 1. For example: RESCOUNT=08	
RESHold	Specifies the code to indicate whether to hold the resource if it is available and in the correct mode. These are the valid values:	
	NO	Do not hold the resource.
	YES	Hold the resource. Another event with a higher dispatch priority can steal the resource.
RESKeep	Specifies the code to indicate whether to keep the resource if the event abends. (This resource can be obtained from a restart/rerun event.) These are the valid values:	
	NO	Do not keep the resource if the event abends. **Note:** If RESMODE (see page 57) is set to SR, then RESASSUME and RESKEEP must be set to NO).
	YES	Keep the resource if the event abends.

Parameter	Description	
RESMode	Specifies the resource mode required by the event. These are the valid modes:	
	ES	(Exclusive/Share) One event has access to this resource exclusively; however, other events can share the resource. ASG recommends that you set the event that has the resource exclusively to EX; you can set other events to ES or SR.
	EX	(Exclusive) One event has access to this resource. The maximum RESCOUNT value for this code is 1.
	SR	Default. Multiple events have access to this resource. The RESASSUME and RESKEEP parameters must be set to NO.
RESName	Specifies the resource name (up to 44 characters long) to be checked for availability before the event is dispatched. The specified resource must be defined in the Zeke database for the event to be scheduled. If you add a resource by specifying only the RESNAME, the other resource parameters will default to these values: RESCOUNT=01 RESMODE=SR RESHOLD=NO RESKEEP=NO RESASSUME=NO If both a RESNAME value and a resource list (indicated by RESOSTART and RESOSTOP) are specified, the RESNAME value is validated and processed, but is replaced by the resource list.	
RESOSTART	For multiple resources, this parameter indicates the beginning of the list of resource definition statements that will replace the ones currently defined for the event. Use the RESOSTOP parameter to indicate the end of the list. You can include only one list of statements for each execution of the EVENT function. If multiple resources are defined for an event, you cannot add or update an individual resource; you must replace the existing list with a new list.	
RESOSTOP	Indicates the end of the list of resource definition statements that will replace the ones currently defined for the event. Use with the RESOSTART parameter (which indicates the beginning of the list).	

Parameter	Description
RESOURCE	Defines a resource. You can define multiple resources by specifying multiple RESOURCE statements (preceded by the RESOSTART parameter and followed by the RESOSTOP parameter). **Note:** The specified resources must be defined in the Zeke database for the event to be scheduled. These are the valid parameters: ASSUME — Specifies whether an event that has RESKEEP set to Y can take the resource from another abended event. These are the valid values: NO — Default. The resource cannot be taken from another abended event. SAME — The resource can be taken only from the same event. YES — The resource can be taken from any event. COUNT — The resource count required by the event. The valid values range from 1 through 999. The default value is 1 (which also is the required setting if RESMODE is set to ES or EX). HOLD — Specifies whether the resource is held (i.e., when it is available and other resources are not available). These are the valid values: NO — Default. Resource is not held. YES — Resource is held. KEEP — Specifies whether this resource is kept if the event abends. These are the valid values: NO — Default. Resource is not kept. This setting is required if RESMODE is set to SH. YES — Resource is kept. MODE — Resource mode. These are the valid values: ES — Exclusive/shared. EX — Exclusive. SR — Default. Shared. NAME — Required. Specifies the resource name (up to 44 characters long).

Parameter	Description
PROCSTEP	Valid for job events only. Specifies the procedure step name after which the resource is released. The default value is blank (i.e., no step name).
STEPNAME	Valid for job events only. Required if STEPNAME is specified. Specifies the step name after which the resource is released. The default value is blank (i.e., no procedure step name).

For example:

```
EVENT ADD JOB NEWJOB ZEKEJCL YES APP TESTAPP GROUP TST
 RESOSTART
  RESOURCE NAME RESO1 MODE SR COUNT 5
  RESOURCE NAME RESO2 MODE EX ASSUME SAME
  RESOURCE NAME RESO3 MODE SR COUNT 4 HOLD YES
 RESOSTOP
```

Event Variable Parameters

These are the valid parameters that you can use with the EVENT function to define event variables:

VARSTART

Indicates the beginning of the list of variables that will replace the ones currently defined for the event or the list of variables to be deleted. Use the VARSTOP parameter to indicate the end of the list. Specify each variable using this format:

```
VARIABLE NAME $name VALUE=value
```

where:

name is a string from two to 16 characters long. It cannot include embedded blanks.

value is a string of mixed case from zero through 64 characters long.

You can include only one list of statements for each execution of the EVENT function.

If multiple variables are defined for an event, you cannot add or update an individual variable; you must replace the existing list with a new list.

For example:

```
EVENT UPDATE 3
 VARSTART
  VARIABLE NAME $ACTDD  VALUE=('SYS2.ACCTING.PROC')
  VARIABLE NAME $RPTYPE VALUE=BASIC
  VARIABLE NAME $BEGIN  VALUE=0001
 VARSTOP
```

VARSTOP

Indicates the end of the list of variables that will replace the ones currently defined for the event or the list of variables to be deleted. Use with the VARSTART parameter (which indicates the beginning of the list).

Parameter	Description
Message Event Parameters	
You can use this parameter with the EVENT function for message events only:	

`ROUTE`	Valid for message events only. Specifies a list of WTO route codes (up to two characters long) to be used when the message is issued. The valid values range from `0` (which removes all route codes) through `16`. A nonzero value adds that code to the list. Parentheses are required.
	Examples:
	This parameter adds a route code 1 to the list:
	`ROUTE=(01)`
	This parameter first resets the list to empty, then adds route codes 11 and 12:
	`ROUTE=(00,11,12)`

Command Event Parameters	
These are the valid parameters that you can use with the EVENT function for command events only:	
`SCOM`*n*	Valid for command events only. Specifies the SCOM command code (where *n* is a value ranging from `1` through `6`), a comma, and the command text (up to 60 characters long). Enclose the line within character string delimiters. For example: `EVENT ADD SCOM SCOM1 (C,F CICSPROD,CENT S...)` **Note:** The SCOM*n* parameters cannot be specified with the SCOMAPPEND or SCOMSTART parameters.
`SCOMAPPEND`	Valid for command events only. Indicates the beginning of SCOM data in the SYSIN JCL to be added to an existing SCOM job as part of the EVENT ADD or EVENT UPDATE process. SCOMAPPEND adds SCOM entries to any that already exist. All SCOM data must begin with the SCOMSTART or SCOMAPPEND statement and end with the SCOMSTOP statement. Each SCOM must begin with a valid SCOM command code, a comma, and the command text (up to 60 characters long). **Note:** SCOMAPPEND cannot be specified with SCOM*n* or SCOMSTART.

Parameter	Description

SCOMSTART	Valid for command events only. Indicates the beginning of SCOM data in the SYSIN JCL to be added for an SCOM event as part of the EVENT ADD or EVENT UPDATE process. SCOMSTART replaces any SCOM entries that already exist.
	
	All SCOM data must begin with the SCOMSTART or SCOMAPPEND statement and end with the SCOMSTOP statement.
	
	Each SCOM must begin with a valid SCOM command code, a comma, and the command text (up to 60 characters long).
	
	SCOMSTART cannot be specified with SCOM*n* or SCOMAPPEND.
	
	This example uses SCOMSTART and SCOMSTOP to add an SCOM event record:
	
	`//ASGADD JOB ,MSGLEVEL=(1,1),CLASS A`
	`//ZUTL EXEC ZEKEUTL,PARM='SUBSYS=SSSI'`
	`//SYSPRINT DD SYSOUT=A`
	`//SYSIN DD *`
	`EVENT ADD SCOM ENAME SCOMEVENT`
	`SCOMSTART`
	`Z,ZID`
	`Z,ZD ALL`
	`C,D A,L`
	`C,D D,T'`
	`SCOMSTOP`
	`//*`

SCOMSTOP	Valid for command events only. Indicates the end of the SYSIN SCOM data to be added to a event as part of the EVENT ADD or EVENT UPDATE process. The SCOM data starts with either the SCOMSTART or SCOMAPPEND statement.

REXX Event Parameters

These are the valid parameters that you can use with the EVENT function for REXX events only:

ARGuments	Valid for REXX events only. Specify the arguments (if any are to be passed to the REXX exec). Enclose all arguments in a single set of character delimiters. The maximum length of the argument is 67 characters.
REXXCLass	Valid for REXX events only. Required to add a REXX event. Specify a valid ECF class in which the exec will run. The valid values range from A through Z and from 0 through 9.
REXXPRIority	Valid for REXX events only. Specify the priority assigned a queued request if all subtasks are busy for the REXXCLASS to which the exec is assigned. The valid values range from 1 through 9 (where 1 is the highest priority). The default value is 5.

Parameter	Description

Task Event Parameters for sFTP

These are the valid parameters that you can use with the EVENT function for sFTP task events:

SFTPCMDS — Required with the ADD function. Indicates the beginning of the list of Zena FTP commands that will replace the ones currently defined for the event.

Note:

The Zena FTP commands that you define in the task event are used to specify the desired FTP operations in the remote Zena FTP task. They are only simplified representations of the commands that will be executed on the FTP host, and not the actual commands.

Use the CMDSSTOP parameter to indicate the end of the list. These are the valid Zena FTP commands:

CHGDIR — Change the local and/or remote directory. This is the command syntax:

```
CHGDIR [CHGLOCAL=N|Y]] [CHGREMOTE=N|Y]] [LOCALDIR='localdir'] [REMOTEDIR='remotedir']
```

where:

localdir specifies the new directory path for the local computer when CHGLOCAL=Y.

remotedir specifies the new directory path for the remote computer when CHGREMOTE=Y.

CHGLOCAL indicates whether to change the local directory path. The valid values are Y (change the local directory) and N (do not change the local directory). The default value is N.

CHGREMOTE indicates whether to change the remote directory path. The valid values are Y (change the remote directory) and N (do not change the remote directory). The default value is N.

DELETE — Delete a file on the remote computer. This is the command syntax.

```
DELETE REMOTEFILE='filename'
```

where:

filename is the filename of the file to be deleted on the remote computer.

GET — Retrieve a file on the remote computer. This is the command syntax.

GETDIR — Retrieve from the remote computer all files in the specified directory and all its subdirectories. This is the command syntax:

```
GETDIR REMOTEDIR='remotedir' [USEASCII=N|Y]
```

Parameter	Description
	where: *remotedir* is the name of the directory on the remote computer for which you want to retrieve its contents. USEASCII indicates the transfer mode. The valid values are Y (use ASCII mode) and N (use binary mode). The default value is N.
LIST	List the filenames on the remote computer. This is the command syntax: LIST FILEMASK='*filemask*' [FILENAME=N|Y] [MODIFIED=N|Y] [SIZE=N|Y] where: *filemask* is the filename or filename mask (e.g., *.CSV) to be listed. FILENAME indicates whether to display the filename. The valid values are Y (display the filename) and N (do not display the filename). The default value is N. MODIFIED indicates whether to request the last file modification date. The valid values are Y (request the date) and N (do not request the date). The default value is N. SIZE indicates whether to request the file size. The valid values are Y (request the size) and N (do not request the size). The default value is N.
MDELETE	Delete one or more files on the remote computer. This is the command syntax: MDELETE REMOTEFILES='*filelist*' [FAILIFNONE=N|Y] *filelist* is a list of filenames or masks that identify the files to be deleted. FAILIFNONE indicates whether to mark the task as failed if no files are deleted. The valid values are Y (fail the task) and N (do not fail the task). The default value is N.
MGET	Retrieve one or more files from the remote computer. This is the command syntax: MGET REMOTEFILES='*filelist*' [USEASCII=N|Y] [FAILIFNONE=[N|Y]

Parameter	Description
	where: *filelist* is a list of filenames or masks that identify the files to be retrieved. FAILIFNONE indicates whether to mark the task as failed if no files are retrieved. The valid values are Y (fail the task) and N (do not fail the task). The default value is N. USEASCII indicates the transfer mode. The valid values are Y (use ASCII mode) and N (use binary mode). The default value is N.
MKDIR	Create a new directory on the remote computer. This is the command syntax: `MKDIR REMOTEDIR='remotedir'` where *remotedir* is the new directory name.
MPUT	Transfer one or more files to a remote computer. This is the command syntax: `MPUT LOCALFILES='filelist' [USEASCII=N|Y] [FAILIFNONE=N|Y]` where: *filelist* is a list of filenames or masks that identify the files to be transferred. FAILIFNONE indicates that the task is marked failed if no files are transferred. The valid values are Y (fail the task) and N (do not fail the task). The default value is N. USEASCII indicates the transfer mode. The valid values are Y (use ASCII mode) and N (use binary mode). The default value is N.
PUT	Send a file to a remote computer. This is the command syntax: `PUT LOCALFILE='localfile' [NEWREMOTE=N|Y] [REMOTEFILE='remotefile'] [USEASCII=N|Y]` where: *localfile* specifies the filename on the local computer. *remotefile* specifies the filename on the remote computer when NEWREMOTE=Y; otherwise, *localfile* is used. NEWREMOTE indicates whether to specify a different filename on the remote computer. The valid values are Y (use a new name) and N (use the local filename). The default value is N. USEASCII indicates the transfer mode. The valid values are Y (use ASCII mode) and N (use binary mode). The default value is N.

Parameter	Description	
	PUTDIR	Transfer all files in the specified directory and all its subdirectories to a remote computer. This is the command syntax: `PUTDIR LOCALDIR='localdir' [USEASCII=N|Y]` where: *localdir* is the name of the local directory for which you want to transfer its contents. USEASCII indicates the transfer mode. The valid values are Y (use ASCII mode) and N (use binary mode). The default value is N.
	QUOTE	Send a QUOTE command to the remote server. This is the command syntax: `QUOTE CMD='text'` where *text* is the argument string for the QUOTE command sent to the remote server.
	RENAME	Rename a file on a remote computer. This is the command syntax: `RENAME OLDFILE='oldfilename' NEWFILE='newfilename'` where: *oldfilename* is the current name of the file. *newfilename* is the new name of the file.
	RMDIR	Remove a directory on the remote system. This is the command syntax: `RMDIR REMOTEDIR='remotedir' [REMOVEALL=N|Y]` where: *remotedir* is the name of the directory to be removed. REMOVEALL indicates whether to remove all data from the directory. The valid values are Y and N. The default value is N (do not remove).
	SITE	Send a SITE command to the remote server. This is the command syntax: `SITE CMD='text'` where *text* is the argument string for the SITE command sent to the remote server.

Parameter	Description
CMDSSTOP	Required when SFTPCMDS is specified. Indicates the end of the list of Zena FTP commands that will replace the ones currently defined for the event. Use the SFTPCMDS parameter to indicate the beginning of the list.
HOSTName	Specifies an override value for the host name defined in the Zeke login record used for the task event. To clear the existing host name, specify an empty string. For example, `HOSTN=()`.
LOGINName	Required with the ADD function. Specify the name of the Zeke login record to use to connect to the remote Zena agent.

This example illustrates the definition of a task event:

```
EVENT ADD TASK SFTP
          ENAME TEST_SFTP_10
          LOGINNAME TCOMUSER
          TARGET ZENASVR1
          HOSTNAME 'my.domain.com'
SFTPCMDS
   CHGDIR CHGLOCAL=Y
          LOCALDIR  'C:\Users\alice\Downloads\Extended'
          CHGREMOTE=Y
          REMOTEDIR '/home/alice/Downloads//filename'
      GET REMOTEFILE 'comma.separated.values.file.xlsx'
          NEWLOCAL=Y
          LOCALFILE 'different.comma.separated.values.file.2.xlsx'
          USEASCII=N
CMDSSTOP
```

Task Event Parameters for Email

These are the valid parameters that you can use with the EVENT function for email task events:

SUBJect	Specifies the email subject. Must be delimited. Required for ADD.
TOADDRess	Specifies the email address of the recipient. Must be delimited. Required for ADD.
IMPORTAnce	Specifies the optional importance of the email. Code NORMAL, LOW, or HIGH. Defaults to NORMAL.
MSGSTART / MSGSTOP	Specifies the optional email message text. Code the desired text lines after the MSGSTART line. End the text lines with MSGSTOP following the last line. Defaults to no message text; the email will have an empty body.

Parameter	Description

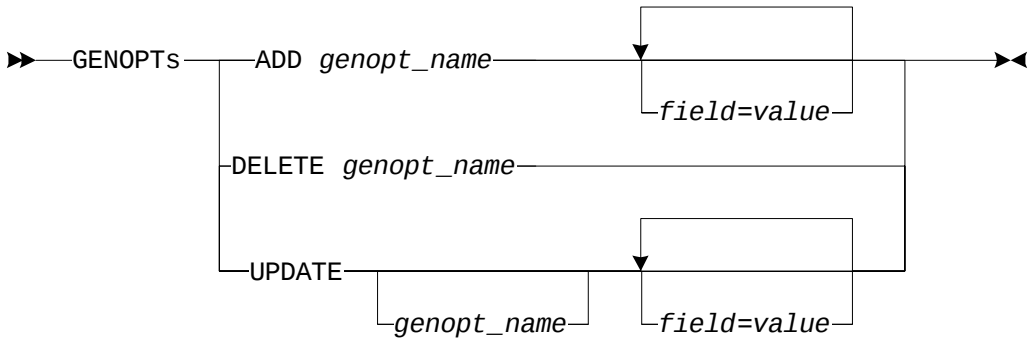
	Example command:
	`EVENT ADD TASK EMAIL`
	`  DESC 'Send email to UK scheduling department'`
	`  ENAME EMAIL_UK`
	`  TARGET UKZENA`
	`  TOADDR 'scheduling@acme.co.uk'`
	`  SUBJ 'Error in daily schedule'`
	`  IMPORTANCE HIGH`
	`  MSGSTART`
	`A Zeke event failed in today's schedule and needs to be researched.`
	`  MSGSTOP`

GENOPTS—Maintain Generation Options

The GENOPTS function enables you to add a new local GENOPT, delete a GENOPT from the Zeke database, or update specific field values for a particular GENOPT.

See Chapter 7, "Generation Options," on page 519 for details on each Zeke generation option. See the *ASG-Zeke Scheduling for z/OS User's Guide* for information on maintaining the generation options using the Zeke online facility.

Syntax



Parameters

Parameter	Description
These are the valid functions:	

ADD	Adds a new local GENOPT with the specified *genopt-name*.
	If no additional parameters are entered, the new GENOPT is added with values set in the default local GENOPT (********). Otherwise, you can specify particular fields and set their values.
	Examples:
	This command creates a new GENOPT named ZEKESYSA with the default values:
	`ADD ZEKESYSA`
	This command creates a new GENOPT named ZEKESYSA with the default values, but changes the value of the DispSel generation option (see page 538) to N:
	`ADD ZEKESYSA DISPSEL=N`

Parameter	Description
DELETE	Deletes the specified GENOPT from the Zeke database. For example: `DELETE ZEKSYSA`
UPDATE	Updates the specified GENOPT. For example, this command updates the PbTrack (problem tracking) field in the GENOPT named ZEKESYSA: `UPDATE ZEKESYSA PBTRACK=Y` If you do not include *genopt-name*, then the currently active local GENOPT is updated.

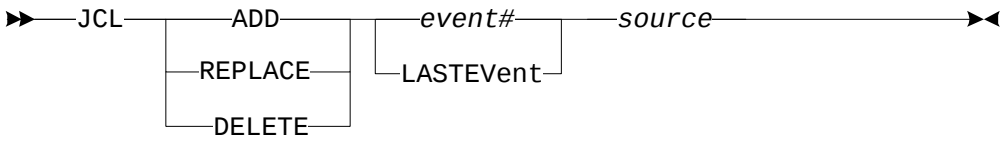
These are the valid parameters that you can use with the GENOPT function:

genopt_name	Specifies the name of the GENOPT (up to eight characters long). For ADD requests, this name must not exist in the Zeke database already. For DELETE and UPDATE requests, this must be the name of an existing GENOPT.
field=value	Specifies the generation option setting, where: *field* is the name of a generation option field. All field names are valid, and you must specify the field name exactly. *value* is a valid value for the specified field (or you can leave it blank to specify no value). For example: `UPDATE *GLOBAL MULTSYS=Y` See Chapter 7, "Generation Options," on page 519 for an alphabetical listing of all Zeke generation option fields. Each field's description also includes the functional category, option type (i.e., local or global), and reload requirement, as well as the valid values.

JCL—Copy JCL into the Database

The JCL function copies JCL from an outside source into the Zeke database. You can use this function to add the JCL to an event, replace existing JCL, or delete JCL.

Syntax



Parameters

You can use these parameters with the JCL function:

Parameter	Description
These are the valid actions for the JCL function:	
ADD	Adds the JCL from an outside source to a Zeke event.
REPLACE	Replaces an event's existing JCL with JCL from an outside source.
DELETE	Deletes the existing JCL from a Zeke event.
These are the valid parameters for the JCL function:	
event#	Specifies the event to be updated.
LASTEVent	Use this parameter instead of the event number to indicate for Zeke to use the event number from the last successful event task (i.e., the last event that was added or updated).
source	Specifies the source of the existing JCL. The DD statement for the source dataset must be specified in the job's JCL or the Zeke procedure. The JCL sources must be defined in the Zeke generation options to enable you to copy JCL from them. These are the valid values:
	BIMLibrary — Specifies the Bim-Edit library from which you want to copy JCL, followed by the parameter BIMMEMBER.
	CONDORMember — Specifies the Condor library member from which you want to copy JCL, and can be followed by the parameters CONDORVER and/or CONDORPASS.

Parameter	Description	
	DDNAME	Required for a partitioned or sequential dataset. Specifies the ddname that contains the member.
	DISK	Specifies the disk dataset from which you want to copy JCL, and can be followed by the parameters DDNAME, UNIT, LRECL, and BLOCK.
	INSTREAM	Indicates that the JCL follows as SYSIN data. All SYSIN JCL data must begin with the JCLSTART statement and end with the JCLSTOP statement. See the JCLSTART parameter on page 47 for more information.
	LIBRARIAN	Specifies the CA Librarian member name from which you want to copy JCL.
	PANVALET	Specifies the CA Panvalet member name from which you want to copy JCL.
	PDS	Specifies the partitioned dataset member name from which you want to copy JCL. The parameter DDNAME is required. For example: `JCL ADD 45 PDS JCL1 DDNAME JCLLIB`
	SEQUENTIAL	Specifies the sequential dataset member name from which you want to copy JCL. The parameter DDNAME is required. For example: `JCL ADD 74 SEQUENTIAL DDNAME SEQIN`
	TAPE	Specifies the tape dataset from which you want to copy JCL, and can be followed by the parameters DDNAME, UNIT, LRECL, and BLOCK.

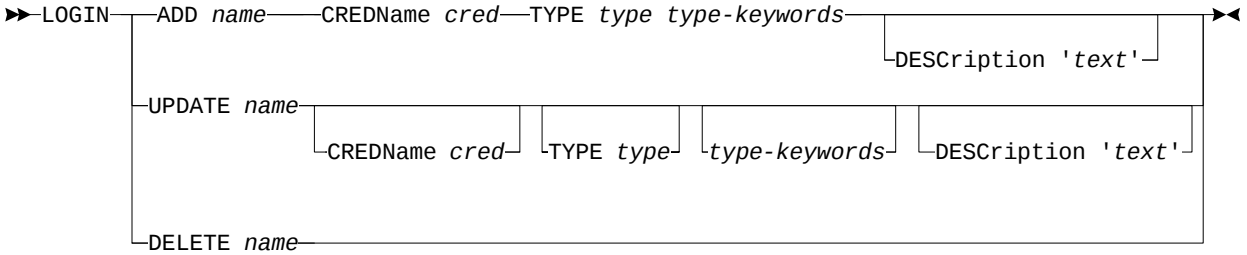
LOGIN—Manage Zeke Logins

The LOGIN function manages Zeke logins. Zeke stores logins in the Zeke database so that they can be assigned to Zeke task events (i.e., tasks that Zeke submits to a remote Zena agent). Zeke requires the appropriate login and user credentials in order to gain access to a remote Zena server.

A Zeke *login* gains authorization to the remote server. A login records specifies parameters required for operating system as well as the associated user credentials. You can assign a particular Zeke login to multiple task events.

See the *ASG-Zeke Scheduling for z/OS User's Guide* for more information on Zeke credentials.

Syntax



Parameters

You can use these parameters with the CREDENTIAL function:

Parameter	Description
These are the valid actions for the LOGIN function:	
ADD	Adds a new Zeke login.
DELETE	Deletes an existing Zeke login from the database.
UPDATE	Updates an existing Zeke login.
These are the valid parameters for the LOGIN function:	
name	Specifies the name (up to 32 characters long) of the login. The name cannot be delimited.

Parameter	Description
CREDName	Required with the ADD function. Specifies the name of an existing Zeke credential (up to 32 characters long) to associate with the login. The value cannot be delimited.
DESCription '*text*'	Description of the login. You can request to display this description when generating reports. The value must be enclosed in delimiters.
TYPE *type*	Required with the ADD function. Specifies the type of login. The value cannot be delimited. These are the valid values: SFTP — Specifies a Secure File Transfer Protocol (SFTP) login. These are the valid keywords: CERTificate '*certpath*' — Specifies the location of the security certificate (for verifying connection to the FTP host) to import to the login record. The value must be enclosed in delimiters. You specify the location using one of these formats: • A fully-qualified pathname: '*/path/to/file*' A pathname must begin with a slash (/), dot (.), or tilde (~) character. • A DD name: 'dd:*ddname*' where *ddname* is the ddname allocated to the utility job that points to the certificate dataset. • A fully-qualified dataset name: 'dsn:*dsname*' where *dsname* is the fully-qualified certificate dataset name. For a PDS, you must include the member name: 'dsn:*pds.dataset*(*member*)' Zeke imports the certificate data when the login record is created or updated. CERTPW '*password*' — Specifies the password associated with the security certificate. The value must be enclosed in delimiters. COMPRess — Indicates whether to perform the transfer using data compression. These are the valid values:

Parameter	Description		
		NO	Default. Do not use data compression.
		YES	Use data compression if the FTP host supports its use.
	HOSTName '*hostname*'	Required with the ADD function. Specifies the host name (up to 255 characters long) where the FTP server resides. The value must be enclosed in delimiters. For example: 'ftp.mydomain.com'	
	HOSTType	Specifies the type of operating system running on the FTP server. These are the valid values (which cannot be delimited): WINDOWS (Default) UNIX MAC	
	PASSIVE	Indicates whether to perform the transfer using FTP passive mode (PASV) if supported by the FTP host. The valid values are No (default) and Yes.	
	PORT	Required with the ADD function. Specifies the port number to be used when connecting to the FTP server. The valid values range from 1 through 32767.	
	PROTOCOL	Required with the ADD function. Specifies the protocol to use for connecting to the FTP host: These are the valid values (which cannot be delimited):	
		EXPLICITTLS	Explicit FTP over Transport Layer Security (TLS).
		IMPLICITTLS	Implicit FTP over TLS or Secure Sockets Layer (SSL).
		SSH2	Default. sFTP using Secure Shell 2 (SSH2).
		UNENCRYPTED	Unencrypted FTP.

Parameter	Description		
	USECERT	Indicates whether to verify the connection to the FTP host using a security certificate. These are the valid values:	
		NO	Default. Do not use a security certificate.
		YES	Use a security certificate. If you specify USECERT=Y, the login must already have a certificate assigned to it or you must also specify the CERTIFICATE parameter on the LOGIN function to import the certificate and import it to the login record.
EMAIL	Specifies a login record that can be used by Email task events to log into an email server. These are the valid keywords:		
	HOSTName	Specifies the address of the SMTP server that is used to send email. The value must be enclosed in delimiters. For example: 'mail.acme.com.ca'	
	PORT	Specifies the SMTP email server's port number. The valid values range from 1 through 32767.	
	FROMADDRess	Specifies the email address to be used as the From: address while sending the email (up to 256 characters log). The value must be enclosed in delimiters. For Example: 'Zeke SYSA <sysa.mainframe@acme.com.ca>'	
	REQUIRELOGIN	Specifies if the email server requires a login. These are the valid values:	
		NO	Default. Do not require a login.
		YES	Allows the email server to check for a login.

Parameter	Description		
	OS	Specifies a general operating system login. These are the valid keywords:	
		ALLOWTOken	Indicates whether to allow the credential's security information to be included in the attributes sent to the FTP host (password information will not be visible). These are the valid values:
			NO — Default. Do not allow the login to use a token.
			YES — Allow the login to use a token. **Note:** Whether you activate this option depends on the requirements of the remote Zena agent (refer to the documentation for the particular Zena agent for more information).
		DOMain '*domainname*'	In the Domain field, specify the domain name to be associated with the login. The value must be enclosed in delimiters.

Sample JCL

This example illustrates the creation of a login:

```
//ZEKELGN  JOB  ,MSGLEVEL=(1,1),CLASS=A
//ZUTL     EXEC ZEKEUTL
//CERTFILE DD   DSN=datset.with.certificate.for.host,DISP=SHR
//SYSIN    DD   *
 LOGIN ADD SPAIN_FTP
       DESC 'Login data for Spain FTP servers'
       HOSTNAME 'ftp.acme.es'
       HOSTTYPE UNIX
       CREDNAME SPAIN_FTP_USER
       TYPE SFTP
       PROTOCOL EXPLICITTLS
       PORT 4752
       COMPRESS=NO
       PASV=NO
       USECERT=YES
       CERTIFICATE 'DD:CERTFILE'
       CERTPW 'password phrase associated with certificate'
/*
```

Sample JCL - EMAIL

This example illustrates how EMAIL parameters are specified:

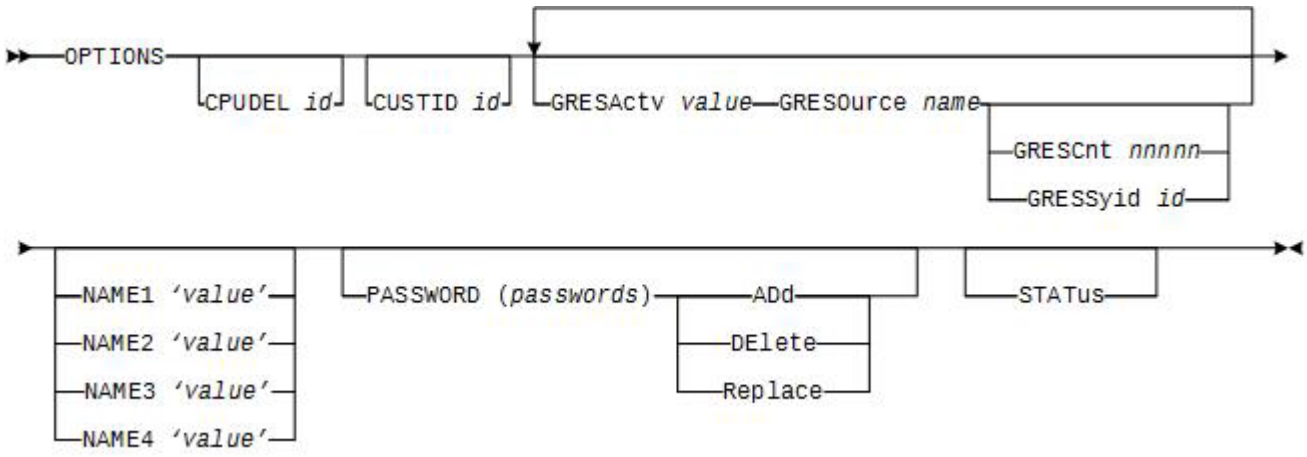
```
LOGIN ADD CA_EMAIL_LOGIN TYPE EMAIL
  DESCRIPTION 'Login for Canada region email servers'
  CREDNAME CA_EMAIL_CREDENTIAL
  HOSTNAME 'mail.acme.com.ca'
  PORT 25
  FROMADDRESS 'Zeke SYSA <sysa.mainframe@acme.com.ca>'
  REQUIRELOGIN Y
```

OPTIONS—Perform Administrative Tasks

The OPTIONS function performs these various functions:

- Establishes and updates the company name and address to be displayed on reports
- Establishes and updates the customer ID
- Prints the Database Status Report

Syntax



Parameters

You can use these parameters with the OPTIONS function:

Parameter	Description
`CPUDEL id`	Deletes an erroneous CPU ID registration shown in a ZD COM display. For example: `OPTIONS CPUDEL ZEKESYSB`
`CUSTID id`	Specifies the customer ID (up to eight characters long). The valid values range from `0` through `9`, and from `A` through `F`), where the fifth position is a value ranging from `0` through `7`. For example: `OPTIONS NAME1 'COMPANY NAME' CUSTID 1AE20F17`

Parameter	Description

Note:

The following GRES*x* parameters have been deprecated. To manage resources, use the RESOURCE command (see "RESOURCE—Maintain Resources" on page 85) instead.

GRESActv *value*	(Deprecated) Indicates whether the resource being defined is active (available). These are the valid values: NO The resource is not available. YES The resource is available.
GRESCnt *nnnnn*	(Deprecated) Specifies the maximum shared count for the resource being defined. This number represents how much of a resource is available to an event. The valid values range from 1 through 65535. The default value is 1.
GRESOurce *name*	(Deprecated) Specifies the name (up to 44 characters long, with no blanks) of the resource being defined. Zeke checks the availability of these resources before the event is dispatched. For example: `OPTIONS GRESOURCE TESTRESOURCENAME GRESSYID SYSTEMA GRESACTV YES GRESCNT 5`
GRESSyid *id*	(Deprecated) Specifies the name of the system that owns the resource. If not specified, this value defaults to GLOBAL to indicate that any system can share this resource. If the event's system ID is assigned to a pool, each resource for that event should be defined as a global resource to ensure proper dispatching. You can specify a resource name more than once with different system IDs.
NAME1 '*value*'	Specifies the company name (up to 40 characters long). Enclose the name within character string delimiters. The company name is printed in the Zeke report headings. For example: `NAME1='MY COMPANY NAME'`
NAME2 NAME3 NAME4	Specifies the company address or other information. You can specify up to three NAME parameters (each one up to 40 characters long). Enclose each line within character string delimiters.
STATus	Prints a status report that lists the amount of used and available space in the Zeke database, as well as information about the contents of the database. For example: `OPTIONS STATUS` See "Sample Database Status Report" on page 80.

Sample Database Status Report

This sample Database Status Report displays the amount of used and available space in the Zeke database for Zeke EMRs and variables:

```
ZEKE 6.20 ZR620000 X320000 - ZEKE UTILITY PROGRAM MAINTENANCE LISTING   MC30  R

 OPTIONS STATUS

           D A T A B A S E   S T A T U S   R E P O R T

DATABASE CREATED: 10/07/2016               ------ EVENT RECORDS -----
    LAST RESTORE: 02/10/2020               ACTIVE:      12  DEAC:         0
     LAST BACKUP: 08/08/2020 15:05:38      USED:        12  UNUSED:  11,812
   RELEASE LEVEL:  6.20                    CAPACITY:  11,824
   ZEKE PGM RUNS:      26                  PERCENT IN USE:    0%

DATABASE BLOCKS:   15,750                  ---- VARIABLE  RECORDS ----
  BLOCKS IN USE:      798                  TOTAL VAR IN USE:      6
    BLOCKS FREE:   14,952
  % BLOCKS USED:      5%

                   ---- EVENT TOTALS BY TYPE ----
                     JOB:       7   SCOM:       0
                    MSGS:       1   WORK:       3
                    PCOM:       0   VCOM:       0
                    ZCOM:       1   REXX:       0
```

This information is displayed on the Database Status Report:

Heading	Description
DATABASE CREATED	Date the Zeke database was created.
LAST RESTORE	Date the Zeke database was last restored.
LAST BACKUP	Date and time the Zeke database was last backed up.
RELEASE LEVEL	Zeke release level.
ZEKE PGM RUNS	Total number of Zeke program executions.
DATABASE BLOCKS	Total number of Zeke database blocks (i.e., the allocated size of the Zeke database).
BLOCKS IN USE	Number of blocks that are being used by Zeke data records.
BLOCKS FREE	Number of free blocks in the Zeke database.
% BLOCKS USED	Percentage of blocks used (i.e., the number of used blocks divided by the total number of blocks available).

Heading	Description
ACTIVE	Number of active EMRs in the Zeke database.
USED	Number of used EMRs in the Zeke database.
CAPACITY	Total capacity of the Zeke database (based on its size) for storing EMRs.
PERCENT IN USE	Percentage of EMRs in use (i.e., the number of used EMRs divided by the capacity).
DEAC	Number of inactive EMRs in the Zeke database.
UNUSED	Number of unused (available) EMRs in the Zeke database.
TOTAL VAR IN USE	Number of variables records in use (i.e., number of variables defined in the Zeke database).
EVENT TOTALS BY TYPE	Total number of events (grouped by event type) owned by this Zeke system.

The report also includes the System Table Status Report, which displays the number of SQRs and WHEN conditions, and the amount of storage required to load the schedule records in memory:

```
           S Y S T E M   T A B L E   S T A T U S   R E P O R T

       +--------------------+--------------------+--------------------+
       |     NUMBER OF      |     NUMBER OF      |                    |
       |  SCHEDULE RECORDS  |  WHEN CONDITIONS   |                    |
       |   IN THE DATABASE  |   FOR THE SQRS     |                    |
       +--------------------+--------------------+   NON WORK CENTER  |
       |           2        |           3        |  <==  EVENTS       |
       +--------------------+--------------------+     WORK CENTER    |
       |           0        |         N/A        |  <==  EVENTS       |
       +--------------------+--------------------+--------------------+
       |   STORAGE NEEDED   |   STORAGE NEEDED   |                    |
       |     FOR LOADING    |      FOR THE       |       TOTAL        |
       |   SQRS TO MEMORY   |   WHEN CONDITIONS  |                    |
       +--------------------+--------------------+--------------------+
       |      10,288 BYTES  |       5,216 BYTES  |      15,504 BYTES  |
       +--------------------+--------------------+--------------------+
Z0247I DATABASE STATUS REPORT COMPLETE
```

This information is displayed on the System Table Status Report:

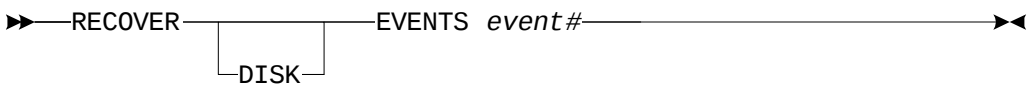
Heading	Description
NUMBER OF SCHEDULE RECORDS IN THE DATABASE	Number of events in the schedule. This field indicates both the number of work center events and the number of non work center events.
STORAGE NEEDED FOR LOADING SQRS TO MEMORY	Number of bytes of storage required to load the SQRs in memory. Work centers are included in this number only if the LoadComm generation option is set to Y.
NUMBER OF WHEN CONDITIONS FOR THE SQRS	Number of WHEN conditions (used to trigger other events) defined in the Zeke database. This field is not applicable to work center events because they do not have WHEN conditions.
STORAGE NEEDED FOR THE WHEN CONDITIONS	Number of bytes of storage required to load the WHEN conditions in memory.
TOTAL	Minimum amount of storage required (as indicated through the SysTbl generation option).

RECOVER—Restore Events from Backup

The RECOVER function enables you to restore individual EMRs from a tape or disk backup file. (Schedule records are not restored.)

The database backup file must have been created by the BACKUP function and must contain a logical backup. You can issue the RECOVER command while Zeke is active.

Syntax



Parameters

You can use these parameters with the RECOVER function:

Parameter	Description
`DISK`	Indicates to restore the events from a disk file. If you do not specify this parameter, then the events are restored from a backup tape (by default).

`EVENTS event#`	Specifies up to 31 event numbers to be restored. Enclose multiple event numbers in parentheses and separate them with commas or spaces. For example:
	`RECOVER DISK EVENTS 1234`
	`RECOVER DISK EVENTS (12 345 6789)`
	Because each event is restored to the Zeke database with its same event number, the event must have been deleted from the database previously.
	Note:
	If an event was deleted using the EVENT DELETE FORCE command (see "EVENT—Add/Maintain Event Master Records" on page 29), then the RECOVER command removes the reserve on the event.

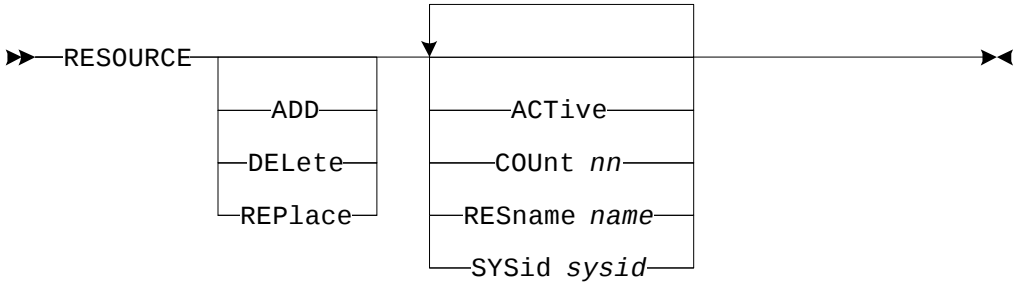
Sample JCL

```
//ZEKERCVR JOB ,MSGLEVEL=(1,1),CLASS=A
//ZEKERS DD DSN=ZEKE.BACKUP,DISP=OLD,
// VOL=(,RETAIN,SER=ZEKETP),UNIT=TAPE,LABEL=(1,SL)
//SYSIN DD *
RECOVER EVENTS (35,400)
/*
```

RESOURCE—Maintain Resources

The RESOURCE command enables you to add, update, and delete logical resources.

Syntax



Parameters

You can use these parameters with the RESOURCE function:

Parameter	Description
`ADD`	Adds a resource to the Zeke database.
`DELete`	Deletes a resource from the Zeke database.
`REPlace`	Replaces the ACTIVE or COUNT parameter values for a resource.
`ACTive`	Indicates whether the resource being defined or updated is active (available). These are the valid values: `N` The resource is not active (available). `Y` Default (when adding a resource). The resource is available. For example: `RESOURCE ADD RES RNO2 SYS TEST ACT YES COU 3`
`COUnt` *`nn`*	Maximum shared count for the resource being added or updated (i.e., number representing how much of a resource is available to an event). The valid values range from `1` through `65535`. The default value is `1` when a new resource is being added. For example: `RESOURCE ADD RES RNO4 SYS PROD ACT NO COU 6`
`RESname` *`name`*	Name of the resource (up to 44 characters long) being defined, updated, or deleted. For example:

Parameter	Description
	`RESOURCE DEL RES RNO1 SYS PROD`

`SYSid` *`sysid`*	Name of the system that owns the resource.
	• If not specified, this value defaults to `GLOBAL` (i.e., any system can share this resource).
	• If the resource is not a GLOBAL resource, then you can specify the same RESNAME more than once with different SYSID values.
	• If the resource is a GLOBAL resource, then only a single resource can exist with the same RESNAME value.
	Examples:
	`RESOURCE REP RES RNO3 SYS TEST ACT NO`
	`RESOURCE ADD RES 'A NAME WITH SPACES' ACT NO COU 99`

RESTORE—Restore a Database Backup

The RESTORE function rebuilds the database from a tape or disk backup file. Use the RESTORE function to recover the contents of a damaged database or to move and/or enlarge the database. The database file must have been created by the ZEKE utility program's BACKUP function (see "BACKUP—Copy the Database or Audit Dataset" on page 8).

Additionally, you can merge a logical backup into an existing database.

These are the two methods to restore the database from a backup:

Logical. This method reorganizes the database, which enables you to restore the database to a larger dataset or merge two databases.

Physical. This method restores the physical portion of the backup to the disk space and creates an exact copy of the backed-up database.

These DD names are required for a database restore:

- ZEKERS specifies the Zeke backup dataset to restore *from*.
- ZEKENEW and ZEKECAT specify the Zeke database to restore *to*. These DD names must point to the same dataset, otherwise the database will not be restored properly. ZEKENEW may be omitted if the MERGE keyword is specified since the database is not initialized in this case.

When the database is being restored, the CREATE function (see "CREATE—Initialize the Database" on page 20) is not necessary because the RESTORE function automatically performs a database creation (unless the MERGE keyword is specified since the database is not created in this case).

Caution! You must completely terminate Zeke before restoring the database. Do not use the RESTORE function to restore an active database. If Zeke is active on any systems sharing the database, stop the sharing systems by issuing the ZKILL COLD command. (Do not use the ZKILL WARM or ZKILL TRACK command.)

Caution! Restore processing will examine the target database to verify there are no Zekes using that database. This processing will fail if the database is damaged or for a new dataset that has not been initialized by the CREATE command (the restore program cannot differentiate these two conditions). For these situations, use the NEWCATLG keyword to bypass the verification. Be sure there are no active Zekes using the database when you use NEWCATLG.

Caution! Do not use the MVS command CANCEL to terminate Zeke unless at the direction of ASG Customer Support.

Note:

Regardless of the value for the EsiActv generation option (see page 546), an external security call always is made to the SAF Security Interface using the resource class of Z$CATAL with a resource name of RESTORE# and ALTER authority. If this class information is *not* defined in your security product, then the SAF action and return code are determined by your security product. If you do not have a security product using SAF, Zeke's internal security is used, which allows the request by default.

Note:

If you have a ZEKE15B user exit in place, then it can override any external security return code depending on how you have defined ZEKE15B.

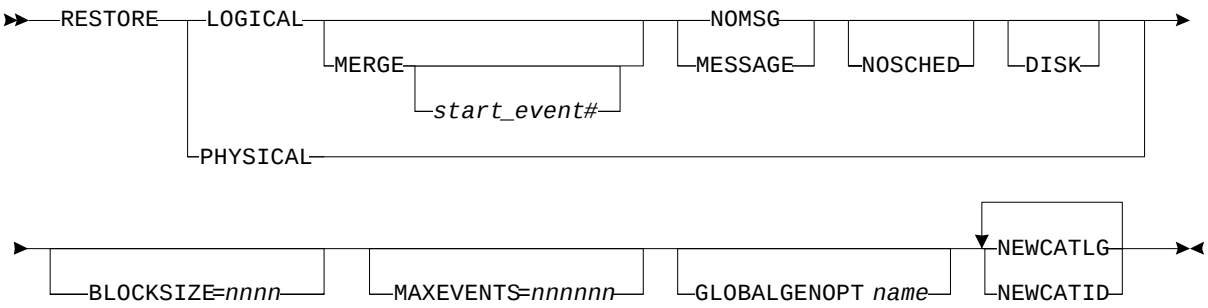
Note:

If you are restoring a Zeke database from a backup containing SQRs that are downloaded to Zeke Agent, then you must either use the RESTORE NOSCHED option, or you must remove the job records from the Zeke Agent that is maintaining SQRs (copies).

ASG recommends that you re-cycle OASIS after performing a database restore.

See the *ASG-Zeke Scheduling for z/OS User's Guide* for more information on creating the Zeke databases (primary and vault).

Syntax



Parameters

You can use these parameters with the RESTORE function:

Parameter	Description
`LOGICAL`	Default. Indicates to reorganize the database, which enables you to restore the database to a larger dataset, or merge two databases. Event numbers are unchanged. Pointers are used to keep event elements grouped together.
`BLOCKSIZE=`*`nnnn`*	Optional. Specifies the block size of the Zeke database. The valid values range from `1024` through `4096` (in 512-byte increments). The default value is `2048` if the `NEWCATLG` parameter is specified. Otherwise, the blocksize of the database allocated to the ZEKECAT DD is used. See "OPTIONS—Perform Administrative Tasks" on page 78 for information on the Database Status Report (which you can use to determine the optimum block size). **Note:** This parameter cannot be used with the MERGE parameter.
`MAXEVENTS=`*`nnnnnn`*	Optional. Specifies the maximum number of events (i.e., the highest event number) that can exist in the Zeke database. The valid values range from `1` through `999999`. The default value is 75 percent of the number of blocks in the database. For example, a database allocated with 100 blocks can have a maximum of 75 events unless this parameter is used to override that value. **Note:** This parameter cannot be used with the MERGE parameter.
`MERGE`	Not valid with the PHYSICAL parameter. Indicates to merge the backup database with the current database. **Note:** MERGE cannot be combined with a database CREATE process.

Parameter	Description

start-event-no	Optional. Not valid with the PHYSICAL parameter. Specifies a number higher than any existing event number in the current Zeke database.
	• Event numbers in the current database are not changed.
	• Duplicate variables, calendars, and security records are skipped.
	• Event numbers specified in OCCURS (REFEVENT) clauses are adjusted accordingly.
	For example, this command merges the backup database with the existing disk database:
	`RESTORE LOGICAL MERGE 1001`
	In the example, event 1 is renumbered as 1001, event 2 becomes 1002, etc.
	When a base EMR record is merged, message Z2E68I indicates the original and merged event numbers. For example:
	`Z2E68I Event Record  000001  restored as 000165`
	If you do not specify a starting event number, these are the effects:
	• Event numbers of the restored events are assigned according to available event numbers in the existing Zeke database. For example:
	`Z2E68I Event Record  000005 restored as 000105`
	• The OCCURS clause REFEVENT is not supported. If the REFEVENT keyword is encountered during the restore, the merge job is aborted with this error:
	`Z2E98E  EVENT (event number): REFEVENT INVALID IN MERGE, RESTORE ABORTED`
	Note:
	To reduce database I/O, be sure that the EDBIndex generation option (see page 545) is set to `Y` before you start the merge job.

`MESSAGE`	Not valid with the PHYSICAL parameter. Indicates to list detailed information in the job output (message Z2E68I) regarding the records that were restored. During a merge, details for duplicate records that are being dropped also are included.
`NOMSG`	Default. Not valid with the PHYSICAL parameter. Indicates *not* to list messages.

Parameter	Description
NOSCHED	Optional. Not valid with the PHYSICAL parameter. Indicates *not* to restore the schedule records. ASG recommends this option for logical restores or merges because the schedule records on a backup tape typically are from the previous day's schedule and are not needed. Use this parameter also for backups that contain downloaded SQRs.
PHYSICAL	Indicates to restore the physical portion of the backup to the disk space; this restores an exact copy of the backed-up database. The target dataset must be the same size and type (CKD/FBA) as the backed up dataset. **Note:** To restore the Zeke database to a larger or a different type dataset, you cannot perform a physical restore; you must perform a logical restore.
DISK	Optional. Indicates to restore the database from a disk file. Otherwise, the database is restored from a backup tape by default.
GLOBALGENOPT *name*	Required if you are restoring from a Zeke 5.6 or earlier database. (Otherwise, the parameter is validated and ignored.) Specifies the name of a GENOPT in the database backup file (if the backup contains multiple GENOPTs) for Zeke to use to create the contents of the *GLOBAL GENOPT for the new database. If the backup has only one GENOPT (which is used by default), you can omit this parameter.
NEWCATID	Indicates that you are restoring or converting a Zeke database from one release to another and plan to run both Zeke releases on the same system (with one of the releases using the original database). As a result of using this option, the newly restored/converted database have a new, unique CATID generated for it by the CREATE process. **Caution!** If you attempt to run multiple releases of Zeke on the same system without this parameter, both databases will have the same CATID and errors could occur. Do not use this parameter if you are using the CATID as part of any ESI resource names. Contact ASG Customer Support for assistance.

Parameter	Description
NEWCATLG	Optional. The default value is 2048 which can be overridden when BLOCKSIZE parameter is specified. Indicates to bypass the verification that there are no active Zeke systems using the target database. Use this option when the dataset has never been initialized by the CREATE function or the database is damaged. **Caution!** This parameter does not consider whether any Zeke systems currently are using the database to be restored. If there are active Zekes, they will not operate correctly until they are restarted after the restore is successfully completed.

Sample JCL

This example restores a backup to a new database:

```
//ZEKEREST JOB  ,MSGLEVEL=(1,1),CLASS=A
//ALLOCDB  EXEC PGM=IEFBR14
//ZEKECAT  DD   DSN=ZEKE.DATABASE,DISP=(NEW,CATLG),
//              UNIT=SYSDA,SPACE=(CYL,(100),,CONTIG),
//              DSORG=PSU
//*
//RESTORE  EXEC ZEKEUTL
//ZEKERS   DD   DSN=ZEKE.BACKUP,DISP=SHR
//ZEKENEW  DD   DSN=ZEKE.DATABASE,DISP=SHR
//ZEKECAT  DD   DSN=ZEKE.DATABASE,DISP=SHR
//SYSIN    DD  *
 RESTORE DISK MESSAGE NEWCATLG
/*
```

This example merges two databases. The ZEKENEW DD is not required because merge processing does not initialize the Zeke database.

```
//ZEKEREST JOB  ,MSGLEVEL=(1,1),CLASS=A
//MERGE    EXEC ZEKEUTL
//ZEKERS   DD   DSN=ZEKE.BACKUP,DISP=SHR
//ZEKECAT  DD   DSN=ZEKE.DATABASE,DISP=SHR
//SYSIN    DD  *
 RESTORE DISK MESSAGE MERGE 1001
/*
```

SCHEDULE—Create the New Schedule

The SCHEDULE function selects the events due for processing and creates the schedule. The SCHEDULE function performs all of these actions:

- Deletes all completed events with a previous run date (RDATE), except for permanent events.
  - — If an event was manually added with a future RDATE, the event is not removed from the schedule until the RDATE is reached (regardless of whether the event has been completed).
  - — If an event was manually added with an RDATE prior to the current date, the event is deleted only if the event's status is Done or if the value for the Retain field in the EMR is `N`.
  - — If an event is a permanent event, you can delete it only by using the ZDELETE command.

  You can delete completed events regardless of their RDATE. To do so, perform a SCHEDULE CLEAR or manually delete the events.

  Note:

  If you run the SCHEDULE ACTIVATE function and include criteria for selecting the events you want to schedule (e.g., event name), the selection parameters you include also can affect which past, completed events are dropped from the schedule. The DropSel generation option (see page 539) determines whether *all* past, completed events are deleted from the schedule when you run the SCHEDULE ACTIVATE function, or only those that match the selection parameters.

- Retains uncompleted events from the previous day's schedule.
- Creates a schedule for a future date. (You can forecast an event schedule and make changes.)
- Analyzes each event defined to the database and determines whether the event will hit during the upcoming schedule period.

Caution! ASG recommends that you *do not* run the SCHEDULE function while a Zeke started task (sharing the database) is down in WARM mode (as the result of a ZKILL WARM command). If you run a Simuload SCHEDULE function while a Zeke started task is down in WARM mode, the Simuload communications records will build up in CSA and is processed when the Zeke started task comes back up.

The typical schedule period is one working day from 00:00 to 47:59 (i.e., 47:59 to select all events that possibly are due); however, you can specify any time range using the SCHEDULE parameter.

For example, a site runs the SCHEDULE function three times a day—from 00:00 to 16:00, from 16:01 to 24:00, and from 24:01 to 47:59. This site has a working day from 8:00 A.M. on the first day to 8:00 A.M. the next day (divided into three shifts). This site's events all are scheduled in the range from 08:00 through 32:00. All times from 00:00 to 47:59 must be covered.

As an alternative, this site could run the SCHEDULE function once per day (before 8:00 A.M.) at the start of the workday. (The third shift processes events scheduled from 24:01 to 32:00 for the previous day.) The SCHEDULE function manages the SQRs in the Zeke database; there is one SQR for each event in the schedule. Existing SQRs could be from previous schedule runs if those events were not completed or disabled. Whether existing SQRs are retained depends on several of the SCHEDULE statement parameters and EMR specifications.

Note:

Zeke provides a user exit that enables you to change various fields in the SQR during the schedule build. See the *ASG-Zeke Scheduling for z/OS Installation Guide* for more information on the ZEKE02OX user exit.

See the *ASG-Zeke Scheduling for z/OS User's Guide* for more information on forecasting and simulating the schedule, creating the Zeke schedule, and setting Zeke to schedule itself.

Forecasting

Using the SCHEDULE command with additional parameters (e.g., DATERANGE, GENERATE, etc.) enables you to produce reports covering a single day's schedule or several days in the future.

Note:

Do not include the ACTIVATE parameter when creating a forecast schedule run; the ACTIVATE parameter updates the database with a new schedule.

Simultaneous Schedule Load (Simuload)

When Simuload (simultaneous schedule load) is in effect, each Zeke system runs its schedule load simultaneously with the schedule creation process of the SCHEDULE function. Each Zeke's schedule is updated (via communication records) as SQRs are updated in the database.

With Simuload, data space creation and schedule loading does not occur separately for each system sharing the Zeke database.

When the SCHEDULE function is complete, one Zeke system will satisfy weak, EOG, extended, and variable WHEN conditions and then the schedule load is completed on all systems. Zeke does not perform a separate schedule load process (and data space creation) for each system sharing the Zeke database.

Simuload occurs automatically if these option settings are in effect:

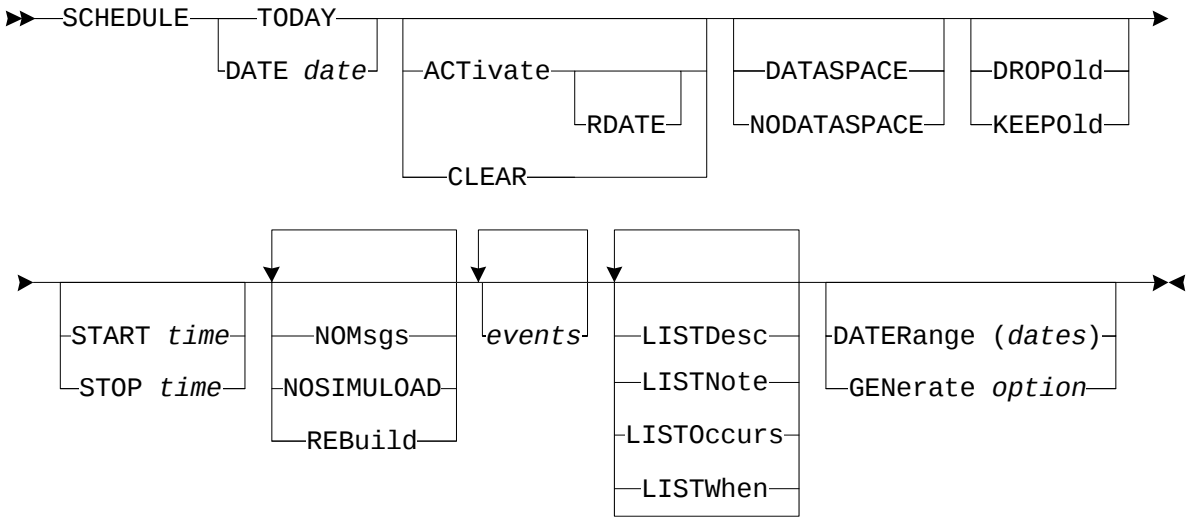
- The Zeke startup option `PLEXENQ=YES`.
- The Zeke startup option `PLEXCOMM=XCFONLY`.

  Or

  The MultSys generation option (see page 560) is set to `N`.

The SCHEDULE function keyword `DATASPACE` is specified (or is the default).

Syntax



Parameters

You can use these parameters with the SCHEDULE function:

Parameter	Description
TODAY	Indicates to uses the current system date as the schedule date. For example: `SCHEDULE TODAY ...`
DATE *date*	Specifies the date (in *mm/dd/yyyy* format) of the schedule run (if other than the current date). For example: `SCHEDULE DATE 01/11/2020 ...` **Note:** If forecasting, do not use the ACTIVATE parameter.
ACTivate	Required to update the database with the new schedule. If omitted, Zeke assumes the schedule run is for forecasting purposes. For example: `SCHEDULE TODAY ACTIVATE` **Caution!** Do not run the SCHEDULE function for a future date when ACTIVATE is specified. If you do so, the current schedule is overwritten.
RDATE	You can use this parameter with the ACTIVATE parameter. Specifies a run date value other than the default value of today. All events added to the schedule will have the run date specified in the statement. For example: `SCHEDULE TODAY ACTIVATE RDATE 12/31/2020` This command updates the schedule to have a run date of 12/31/2020. **Note:** If you use this parameter to assign a future run date to run an extra workload, in addition to the regular workload for that day, consider how you want cross-triggering to occur between the two loads. The triggering behavior depends on the setting of the TrigDt generation option (see page 588). For example, suppose your regular workload has a schedule date of 10/31 and a run date of 10/31, and your extra workload has a schedule date of 10/30 and a run date of 10/31. If the TrigDt generation option is set to R, then cross-triggering will occur between the two schedules since the run dates are the same; if TrigDt is set to S, then cross-triggering will *not* occur since the schedule dates are different.

Parameter	Description

CLEAR	Removes all schedule records with the specified criteria (except for permanent events). You can specify schedule records by group ID, application ID, and user ID. For example, this command removes the schedule records with an application ID of AAC, a group ID of DEP and a user ID of PAYRL:
	`SCHEDULE TODAY CLEAR APP=AAC GROUP=DEP USER=PAYRL`
	Notes:
	• If a schedule record does not satisfy all the criteria specified in the statement, the schedule record remains in the schedule.
	• Any logical resources that were acquired for the scheduled event are released before the event is purged.
	To clear all (non-permanent) schedule records, issue this command:
	`SCHEDULE TODAY CLEAR`

DATASPACE	Creates a copy of the Zeke database in a temporary data space. Zeke will use the data space to read the EMRs for creating the schedule and the SQRs for event processing (instead of using the database). If you always want to use a data space for creating the schedule, set the DSPBatch generation option to Y (see page 541). If you include the DATASPACE keyword, a temporary data space is created when the SCHEDULE function starts. (If you also include the LISTWHEN keyword, a data space is created again before any schedule reports are generated if LISTWHEN is included). The use of data spaces by the ZEKE utility program is managed so that only one data space is required to complete the entire schedule creation process: • As long as the SCHEDULE parameter is used in the same execution of the ZEKE batch utility, you can reuse a data space (created by a database BACKUP function or a REPORT function) for schedule creation. • Multiple SCHEDULE commands (for multiple Zeke systems) in the same execution of the ZEKE utility program can use the same data space. **Note:** Using this parameter can significantly reduce database I/O, the duration of database enqueues, and the time required for a schedule build.
NODATASPACE	If the DSPBatch generation option (see page 541) is set to Y, the ZEKE utility program will use a data space when creating the schedule. Use this parameter to override this setting for a particular execution (so that a data space is not used).

Parameter	Description
DROPOld	Deletes SQRs from a prior day regardless of completion status. If omitted, the prior day's SQRs are deleted only if flagged as completed or disabled or if the Retain field is set to N in the EMR. For example: `SCHEDULE TODAY ACTIVATE DROPOLD`
KEEPOld	Retains completed SQRs from a prior day's schedule. If omitted, the prior day's SQRs are deleted if they are flagged as completed or disabled. For example: `SCHEDULE TODAY ACTIVATE KEEPOLD ...`
START	Specifies the starting time for this schedule run, in *hh:mm* format. The valid values range from 00:00 (default) through 47:59. This parameter selects all events with a SCHEDULE time equal to or greater than the specified time. Specifying a start time will prevent the schedule from selecting events with no schedule time. For example, this command places events in the schedule that have a start time of 08:00 or later. If you want to have events added to the schedule with start times from 00:00 through 07:59, then a separate SCHEDULE TODAY statement is necessary: `SCHEDULE TODAY ACTIVATE START 08:00` **Note:** Use this parameter only if the schedule day is divided into separately scheduled segments or time ranges.
STOP	Specifies the ending time for this schedule run *hh:mm* format. The valid values range from 00:01 to 47:59. This parameter selects all events with a SCHEDULE time equal to or earlier than the specified time. For example, this command places events in the schedule that have a start time from 08:01 through 16:00. If you want to have events added to the schedule with start times outside this range, then a separate SCHEDULE TODAY statement is necessary: `SCHEDULE TODAY ACTIVATE START 08:01 STOP 16:00` **Note:** Use this parameter only if the schedule day is divided into separately scheduled segments or time ranges.
NOMsgs	Suppresses message Z02A1I during the schedule run. This message gives the disposition of each schedule record as it is processed. The *ASG-Zeke Scheduling Messages and Codes Guide* includes a list all dispositions with explanations.
NOSIMULOAD	Disables automatic Simuload in a Zekeplex. See "Simultaneous Schedule Load (Simuload)" on page 94 for more information.

Parameter	Description
REBuild	Rebuilds existing SQRs for the current schedule date, as long as they are still active (i.e., have not been dispatched). The rebuilt SQR reflects the updated EMR information and resets all WHEN conditions. This parameter is useful when the EMRs are updated after the SQRs are created. For example: `SCHEDULE TODAY ACTIVATE REBUILD` **Note:** ASG recommends that you rebuild individual schedule records by using the ZADD REBUILD command.

These parameters are valid with the SCHEDULE function to select events:

APPLication	Selects events with the specified application ID. For example:
	`SCHEDULE TODAY ACTIVATE APP PAY`
	You can specify generic selection criteria (see "Generic Selection Criteria" on page 5).
	You can schedule up to 20 application IDs at one time. To do so, enclose the set of IDs in parentheses and separate each ID with a comma. For example:
	`SCHEDULE TODAY ACT APP (APP1,APP2,APP3, APP4,APP5,APP6,APP7,APP8,APP9)`

DRL	Selects events with the specified disaster recovery level (DRL). You must specify the value within delimiters. For example, this command selects only the disaster recovery levels 1 through 4, and 7: `SCHEDULE TODAY ACTIVATE DRL (1-4,7)`
ENAME	Selects events with the specified event name (up to 12 characters long). For example: `SCHEDULE TODAY ACT ENAME ABCDEFGHIJK` You can specify generic selection criteria (see "Generic Selection Criteria" on page 5). You can schedule multiple event names (up to 20) at one time. To do so, enclose the set of names in parentheses and separate each name with a comma.

Parameter	Description

GROupid	Selects events with the specified group ID (up to three characters long). For example:
	`SCHEDULE TODAY ACTIVATE GROUP SAL`
	You can specify generic selection criteria (see "Generic Selection Criteria" on page 5).
	You can schedule multiple group IDs (up to 20) at one time. To do so, enclose the set of IDs in parentheses and separate each ID with a comma. For example:
	`SCHEDULE TODAY ACT GR (GR1,GR2, GR3,GR4,GR5,GR6,GR7,GR8,GR9)`

JOB	Selects job events with the specified jobname (up to 30 characters long). Jobnames are case-sensitive. For example: `SCHEDULE TODAY ACT JOB JOB12345` You can specify generic selection criteria (see "Generic Selection Criteria" on page 5). You can schedule multiple jobnames (up to 20) at one time. To do so, enclose the set of names in parentheses and separate each name with a comma.
SYStemid	Selects only events that belong to the specified system ID. You can specify generic selection criteria (see "Generic Selection Criteria" on page 5). You can specify up to 20 systems. If omitted, the schedule is run for all systems defined in the Zeke database.

USERid	Selects events with the specified user ID (up to eight characters long). User IDs are case-sensitive. For example:
	`SCHEDULE TODAY ACTIVATE USER PAYROLL`
	You can specify generic selection criteria (see "Generic Selection Criteria" on page 5).
	You can schedule multiple user IDs (up to 20) at one time. To do so, enclose the set of IDs in parentheses and separate each ID with a comma. For example:
	`SCHEDULE TODAY ACTIVATE US (USER1, USER2,USER3,USER4,USER5,USER6,USER7)`

Parameter	Description
These parameters are valid with the SCHEDULE function to specify reporting options:	
`LISTDesc`	Prints the event's description on the schedule report.
`LISTNote`	Prints the event's note information on the schedule report. For example: `SCHEDULE TODAY ACTIVATE LISTNOTE`
`LISTOccurs`	Prints the event's OCCURS clause on the schedule report.
`LISTWhen`	Prints the event's WHEN condition on the schedule report. For example: `SCHEDULE TODAY ACT LISTOCCURS LISTWHEN` **Caution!** Because LISTWHEN causes an additional data space to be created when the SCHEDULE function runs, ASG recommends you use this option only when necessary.
These parameters are valid with the SCHEDULE function to specify forecasting options:	
`DATERange (`*`dates`*`)`	Specifies up to five pairs of date ranges for a forecast schedule run. Enclose the values in parentheses and separate them by commas or spaces. If an ending date is not specified for the last pair, then the same date as the starting date is assumed. For example, this command produces a forecast for January 15, 2020 through January 25, 2020, and a forecast for February 14, 2020: `SCHEDULE DATERANGE (01/15/2020, 01/25/2020 02/14/2020)`
`GENerate` *`option`*	Valid only with the DATERANGE parameter. This parameter specifies how to produce the forecast schedule run reports. These are the valid options: `RANGE` Default. Produces one report for each date pair. `DAY` Produces one report for each date in a range. `ONE` Produces one report for all the days.

Parameter	Description

	Examples:
	This command produces two sets of reports (one for each date pair):
	`SCHEDULE DATERANGE (10/01/2020,10/03/2020,10/20/2020, 10/22/2020) GENERATE RANGE DATERANGE (10/01/2020, 10/03/2020,10/20/2020,10/22/2020)`
	This command produces six reports (one for each date forecast):
	`SCHEDULE DATERANGE (10/01/2020,10/03/2020,10/20/2020, 10/22/2020) GENERATE DAY`
	This command produces one report:
	`SCHEDULE DATERANGE (10/01/2020,10/03/2020,10/20/2020, 10/22/2020) GENERATE ONE`

Sample Output

This example shows the output generated after the SCHEDULE function runs successfully:

```
Z0298I SCHEDULING RUN SUCCESSFULLY COMPLETED
Z02A6I THE NUMBER OF SQRS CALCULATED FOR THIS SCHEDULE RUN IS      60
Z02A7I THE NUMBER OF WHEN SEGMENTS CALCULATED FOR THIS SCHEDULE RUN IS      50
Z02F3I Number of serial mismatches:       0
```

OVERRIDE—Select Events

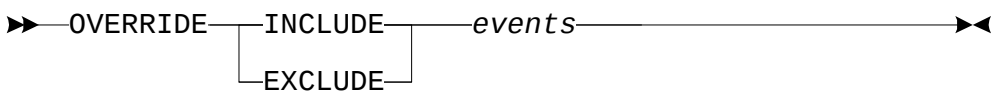
The OVERRIDE subfunction of the SCHEDULE function enables you to include or exclude events from the schedule, regardless of their OCCURS clauses. An event specified in an OVERRIDE INCLUDE command is included even if the schedule run time range does not include the event's schedule time.

For example, if the schedule function runs with `START 08:00` and `STOP 12:00` and event 4 has a schedule time of 16:00, normally the event is not selected because the schedule time is out of the range. However, if event 4 is specified in an OVERRIDE INCLUDE command, the event is selected, regardless of the schedule time.

Issue the OVERRIDE command before the SCHEDULE command or commands. It remains in effect for the duration of the ZEKE utility program.

See the *ASG-Zeke Scheduling for z/OS User's Guide* for more information on creating the Zeke schedule.

Syntax



Parameters

You can use these parameters with the OVERRIDE function:

Parameter	Description
`INCLUDE`	Indicates to select the specified events.
`EXCLUDE`	Indicates *not to* select the specified events.
`events`	Specifies up to 62 individual event numbers and/or 30 event ranges for each OVERRIDE command. Enclose multiple event numbers within parentheses and separate by commas or spaces. Separate the beginning event number of a range and the ending number with a hyphen (`-`). The beginning and ending event numbers are included in the range.

Examples

This command includes event 2:

```
OVERRIDE INCLUDE 2
```

This command includes events 3, 15 through 30, and 45 through 60:

```
OVERRIDE INCLUDE (3,15-30,45-60)
```

This command excludes events 4 and 7:

```
OVERRIDE EXCLUDE (4 7)
```

This command includes event 10 and excludes events 5 and 7:

```
OVERRIDE INCLUDE 10 EXCLUDE (5 7)
```

Sample JCL

Note:

Typically, you issue only one OVERRIDE command per batch program. If a batch program issues multiple OVERRIDE commands, each command adds to, and subtracts from, the selections that resulted from the previous OVERRIDE commands.

For example:

```
//SCH   EXEC ZEKEUTL
//SYSIN DD *
  OVERRIDE INCLUDE 5             Includes event 5.
  SCHEDULE TODAY ACTIVATE

  OVERRIDE INCLUDE (10 11)       Includes events 5, 10, and 11.
  SCHEDULE DATE 12/01/2017

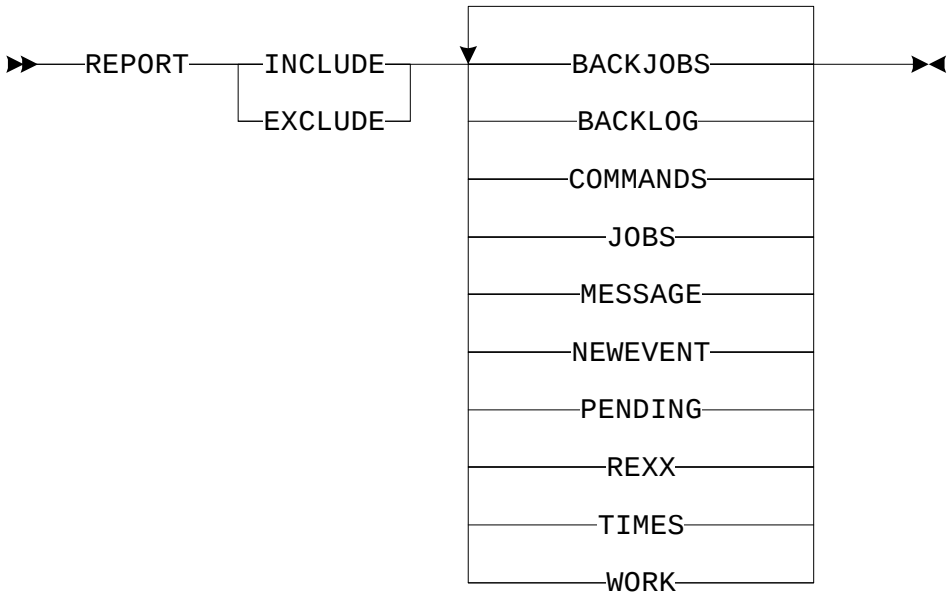
  OVERRIDE EXCLUDE (30-40)       Excludes events 30 through 40.
  SCHEDULE DATE 12/31/2017
/*
```

REPORT—Generate Schedule Reports

This section explains how to generate schedule reports using the REPORT subfunction of the SCHEDULE function. By default, the SCHEDULE function prints ten reports. You also can select only specific information to report and print (e.g., if you are testing an OCCURS clause for a new event).

See the *ASG-Zeke Scheduling for z/OS User's Guide* for more information on creating the Zeke schedule.

Syntax



Parameters

You can use these parameters to generate a schedule report using the REPORT function:

Parameter	Description
INCLUDE	Indicates which types of information to include on the report.
EXCLUDE	Indicates which types of information to exclude from the report.
BACKJOBS	Specifies the BACKJOBS report (which includes job events not completed from the prior day).
BACKLOG	Specifies the BACKLOG report (which includes events not completed from the prior day).
COMMANDS	Specifies the COMMANDS report (which is the schedule of command events).
JOBS	Specifies the JOBS report (which is the schedule of jobs events).
MESSAGE	Specifies the MESSAGE report (which is the schedule of message events).
NEWEVENT	Specifies the NEWEVENT report (which includes events added by the SCHEDULE function).
PENDING	Specifies the PENDING report (which includes events pending or abended).

Parameter	Description
REXX	Specifies the REXX report (which is the schedule of REXX events).
TIMES	Specifies the TIMES report, which is a schedule listing for a specified time range: • If the SCHEDULE function specifies the START and STOP parameters (see page 98) to indicate the starting and ending times for the schedule run, then the TIMES report is based on that time range. • If the SCHEDULE function does not specify a time range, then default time range (00:00 through 47:59) is reported.
WORK	Specifies the WORK report (which is the schedule of work center events).

Sample JCL

Issue the REPORT command before issuing the SCHEDULE command. Typically, you issue only one REPORT command when using the ZEKE utility program. This report specification remains in effect for the duration of any ZEKE utility program activity. If you enter more than one REPORT command while the ZEKE utility program is active, each REPORT command adds to, and subtracts from, the previous REPORT command.

For example:

```
//SCH   EXEC ZEKEUTL
//SYSIN DD *
  SCHEDULE TODAY ACTIVATE      Prints all reports

  REPORT EXCLUDE NEWEVENT JOBS Prints all reports except NEWEVENT and JOBS
  SCHEDULE DATE 12/01/

  REPORT INCLUDE NEWEVENT      Adds NEWEVENT; prints all reports except JOBS
  SCHEDULE DATE 12/31/
/*
```

SIMULATE—Simulating the Schedule

The SIMULATE function enables you to simulate the Zeke schedule run and report the results as if the schedule had actually run. Creating a simulation enables you to test the run time of a schedule based on various factors (e.g., newly-added jobs, reduced or increased resources, etc.).

The Zeke SZEKINS0 library contains sample member ZEKESIM for your reference.

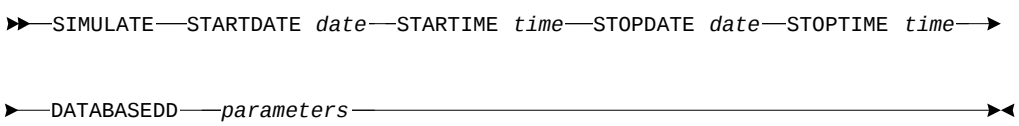
Note:

IBM Workload Manager scheduling environments are not considered during simulation.

Caution! *Do not* run the simulation function against the production database. Doing so will destroy the production database. Run the simulation only against a database copied for that purpose. No other Zeke system should be running against the same database as a simulation.

See the *ASG-Zeke Scheduling for z/OS User's Guide* for more information on simulating the schedule.

Syntax



Parameters

You use these parameters with the SIMULATE function:

Parameter	Description
STARTDATE	Specify the simulation start date, in either *mm/dd/yyyy* or *yyyyddd* format. For example: `SIMULATE STARTDATE 01/01/2020`
STARTTIME	Specify the simulation start time, in *hh:mm* format. The valid values range from `00:00` to `24:00`. For example: `SIMULATE STARTTIME 00:00`
STOPDATE	Specify the simulation stop date, in either *mm/dd/yyyy* or *yyyyddd* format. For example: `SIMULATE STOPDATE 01/02/2020`
STOPTIME	Specify the simulation stop time, in *hh:mm* format. The valid values range from `00:00` through `23:59`. For example: `SIMULATE STOPTIME 23:59`

Parameter	Description
DATABASEDD	Specify the simulation database ddname (up to nine characters long). For example: `SIMULATE DATABASEDD OUTCAT` **Note:** If you are creating a database copy for the simulation, you must specify the same value for DATABASEDD as for the COPY subparameter TODD. If both DATABASEDD and TODD are set to DATASPACE, the simulation database is copied to a data space. See "Running Simulation from a Data Space" on page 113 for more information.
SYSTEM	Specify the name of the system you want to simulate. For example: `SIMULATE SYSTEM MVSSPA` **Note:** If the simulation run does not schedule any events with this system name, then Zeke does not simulate any event dispatching. You can simulate only one system. If you specify multiple system parameters, Zeke ignores all except the last one.
INITIATORS	Specify the number of available initiators. The valid values range from 1 through 230. For example: `SIMULATE INITIATORS 10` **Note:** If more initiators are specified than are defined in the GENSYS record for the system, then simulation generates its own initiator names.
TAPEDRIVES	Specify the number of available tape drives. The valid values range from 1 through 999. For example: `SIMULATE TAPEDRIVES 5`

SATISFY	Specify the conditions to be satisfied automatically. Unless you use this parameter and specify a condition, the simulation process assumes that there are no conditions to be satisfied automatically.
	You can specify one or more of these conditions:
	`ALL*` `AT***`
	`AEOP` `OPOK`
	`AEOS` `UOCCEXIT`
	`BOP` `VAR`
	`DSN` `WHEN**`
	`EOP` `WRKCNTR`
	`EOS`

Parameter	Description
	* Satisfies all of the conditions. ** Excludes satisfaction of OPOK, UOCCEXIT, VAR, and WRKCNTR. *** Satisfies all WHEN conditions with remote prerequisites. For example: `SIMULATE SATISFY OPOK`
`SCHEDRUN`	Specify when to run the SCHEDULE function. These are the valid values: `DAILY` Default. Run the schedule every 24 hours during simulation. `YES` Run the schedule once (at the beginning of simulation). `OFF` Do not run the SCHEDULE function.
`SCHEDCLR`	Specify whether to run the schedule clear function. These are the valid values: `YES` Default. Run the schedule clear once (at the beginning of simulation). `OFF` Do not run the schedule clear function.

Note:

To simulate your current schedule, create a simulation copy against the existing database and its schedule records (i.e., SIMULATE COPY) and set the SCHEDRUN and SCHEDCLR parameters to `OFF`.

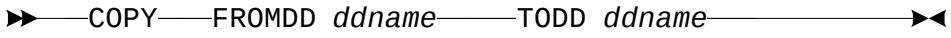
`APPLICATION`	Valid only for SCHEDULE RUN or SCHEDULE CLEAR. Selects events with the specified application ID. If omitted, events with any application ID are selected.
`GROUPID`	Valid only for SCHEDULE RUN or SCHEDULE CLEAR. Selects events with the specified group ID. If omitted, events with any group ID are selected.
`USERID`	Valid only for SCHEDULE RUN or SCHEDULE CLEAR. Selects events with the specified user ID. If omitted, events with any user ID are selected.
`SYSTEMID`	Selects events for the specified system ID. If omitted, the schedule is run for all systems.

Parameter	Description
PERMFREQ	Specify the frequency (in *hh:mm* format) for all permanent events in the schedule that have a frequency of 00:00. The valid values range from 00:00 through 48:00. If you do not specify a value, the default value is 24:00. Permanent events with a frequency greater than 00:00 are not modified. When a simulation is started, the SIMULATE function continuously dispatches permanent events that have a frequency of 00:00. Use the PERMFREQ parameter to prevent this occurrence. For example: `SIMULATE PERMFREQ (12:00)`

COPY—Create a Simulation Database

The COPY function creates a simulation database by copying the specified existing Zeke database. To create a database copy, Zeke must be active and using the database to be copied. The simulation also must be run using the same subsystem as Zeke.

Syntax



Parameters

You use these required parameters to generate a create a simulation database using the COPY function:

Command	Description
FROMDD	Required. Specify the source database ddname the simulation function is copying.
TODD	Required. Specify the destination database ddname being copied by the simulation function. For example: `COPY FROMDD=INCAT TODD=OUTCAT` The TODD parameter must have the same value as DATABASEDD (see page 108). If both values are DATASPACE, the simulation database is copied to a data space. See "Running Simulation from a Data Space" on page 113 for more information.

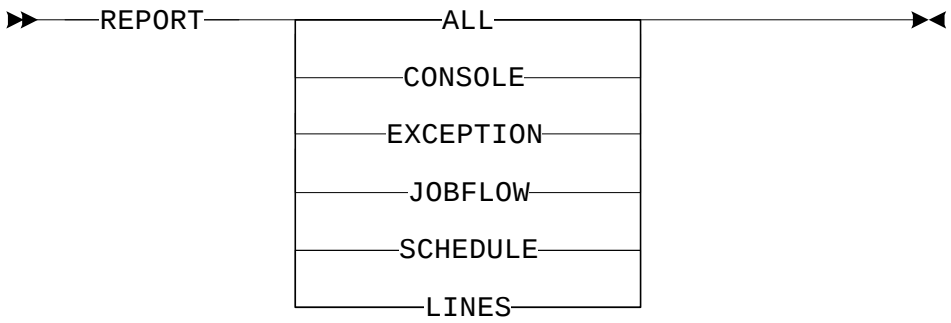
REPORT—Generate Simulation Reports

This section explains how to generate simulation reports using the REPORT subfunction of the SIMULATE function.

To print reports from a previous simulation run, ensure that the ZKSMLOG dataset was saved from the previous run. Then, specify only REPORT parameters in the SYSIN control statements and point the ZKSMLOG DD to the saved log. You do not need to rerun the simulation.

See the *ASG-Zeke Scheduling for z/OS User's Guide* for more information on forecasting and simulating the schedule.

Syntax



Parameters

You use these parameters to generate a simulation report using the REPORT function:

Command	Description
ALL	Prints all simulation reports. For example: `REPORT ALL`
CONSOLE	Prints the console log from the simulation run.
EXCEPTION	Prints the exception report from the simulation run.
JOBFLOW	Prints the initiator job flow report from the simulation run.
SCHEDULE	Prints the simulation schedule report.
LINES	Number of lines per page for the report. The default value is `60`.

Sample JCL

This example illustrates the control statements for executing the Zeke simulation function:

```
...
  COPY       FROMDD=INCAT TODD=OUTCAT
  SIMULATE   STARTDATE 01/01/  STARTTIME 23:00
             STOPDATE  01/02/  STOPTIME  22:59
             DATABASEDD OUTCAT
             SATISFY ALL
             INITIATORS 10
             SYSTEM MVSSPA
             TAPEDRIVES 5
  REPORT ALL
...
```

Note:

Simulation requires its own, unique SYSMDUMP dataset. See the troubleshooting appendix in the *ASG-Zeke Scheduling for z/OS Installation Guide* for details on enabling IPCS readable dumps.

It is not necessary to repeat the COPY function after you have copied the dataset. You can change values for any of the parameters and run a different schedule simulation using the same dataset. For example:

```
...
  SIMULATE   STARTDATE 01/01/  STARTTIME 23:00
             STOPDATE  01/02/  STOPTIME  22:59
             DATABASEDD OUTCAT
             SATISFY ALL
             INITIATORS 5
             SYSTEM MVSSPA
             TAPEDRIVES 3
  REPORT ALL
```

Note:

If you are not performing a COPY, Zeke does not need to be active.

If you are only printing reports, the ZKSMLOG DD provides the data for the reports (and was generated by a previous simulation run). For example:

```
...
//ZKSMLOG  DD  DSN=ZEKE.REPORT.LOG,DISP=SHR,
//             UNIT=3380,VOL=SER=ASWDLB,
//             DCB=(LRECL=256,BLKSIZE=5124,RECFM=VB)
...
  REPORT ALL
//
```

Running Simulation from a Data Space

You can run simulation where the simulation copy of the database is created in a z/OS data space instead of on DASD. This method significantly reduces in the running time of the job and the amount of I/O it performs; however, the data space containing the simulation database is discarded at the end of the job. If you want to save the schedule, or any other database updates made by the simulation run, you must run simulation with its database on DASD.

To place the simulation database in a data space, use the special ddname DATASPACE in the TODD parameter of the COPY command and the DATABASEDD parameter of the SIMULATE command. For example:

```
...
COPY     FROMDD=INCAT  TODD=DATASPACE
SIMULATE   STARTDATE 04/19/ STARTTIME  05:00
           STOPDATE 04/20/ STOPTIME  05:00
           SATISFY ALL
           INITIATORS 7
           SYSTEM SYSNAME
           DATABASEDD DATASPACE
 REPORT ALL
...
```

VAULT—Control the Vault Dataset

The VAULT function disables electronic vaulting while Zeke is down.

Note:

See also “Disabling Electronic Vaulting” on page 310 for information on performing this function using the ZDISABLE command.

If the vault is disabled via the ZEKE utility program, then remove the ZEKEVLT ddname from the Zeke started task JCL before starting Zeke again. Failure to do so will result in vault initialization.

See the *ASG-Zeke Scheduling for z/OS User's Guide* for more information on creating the Zeke databases (primary and vault) and recovery using electronic vaulting.

Syntax



2 ZEKEXUTL Import/Export Utility

This chapter explains how to use the ZEKEXUTL utility program to import and/or export database records. It contains these topics:

Topic	Page
Importing and Exporting Database Records	116
Implementation Requirements	116
Allocate XML Datasets	116
Install RI Server	119
Configure the Zeke Server	119
Define OMVS Segment in User Profile	120
Executing the ZEKEXUTL Utility	120
LE Runtime Options	122
Log Messages	122
Control Statements	122
EXPORT—Export Database Records to a File	127
EXPORT Examples	133
IMPORT—Import Records from a File or Database	134
IMPORT Example	137
FILTER—Filter Imported/Exported Records	138
Filtering Rules	141
CHANGE—Change Imported Record Values	144
CHANGE Examples	149
Multiple Change Rule Processing	150
REPORT—Creating Import/Export Reports	152
Report Examples	154
Sample Import and Export Jobs	160
Sample EMR Export Jobs	160
Sample Calendar Export Job	162
Sample Calendar Import Job	163
Sample Mass Record Change Jobs	164
Moving from a Test to a Production System	166

Importing and Exporting Database Records

You use the import/export utility program ZEKEXUTL to perform these procedures:

- Export these types of Zeke database records as XML data:
  - — Event master records (EMR)
  - — Variable records (VAR)
  - — Calendar records (CAL)

  (See "EXPORT—Export Database Records to a File" on page 127.)
- Import event, variable, and calendar XML records into the Zeke database.

  (See "IMPORT—Import Records from a File or Database" on page 134.)
- Change key values in the records being imported or exported.

  (See "CHANGE—Change Imported Record Values" on page 144.)

These two types of control statements are used with the IMPORT and EXPORT functions:

- *Filtering* control statements enable you to select which records to import or export. Filtering statements have a default value and include at least one IF statement for setting selection criteria.
- *Change* control statements enable you to change fields within the records being imported or exported. Change statements specify whether the entire record will be exported, or only key fields and any changed fields. IF/THEN statements can be used to set selection criteria for the records to be changed.

See "Control Statements" on page 122 for more information, including general formatting rules and guidelines.

Implementation Requirements

This section outlines the requirements for implementing the ZEKEXUTL utility program.

Allocate XML Datasets

XML datasets must be allocated with enough space for all exported records. Datasets require a sufficient record length to contain a complete record element, its attributes, and the variable length values for those attributes.

Note:

If there is insufficient LRECL for the output dataset when the utility attempts to write a record to the XML file, then the ZEKEXUTL utility program issues an error message and terminates.

Space Considerations

ASG recommends include these specifications for allocating datasets:

- Variable length record format.
- Block size equal to one-half of a DASD track. This value may range from 1,028 to 32,767. Exact block size depends on the DASD device on which the dataset is allocated.
- Minimum logical record length (LRECL) of four bytes less than the block size value.

For example:

```
DCB=(LRECL=27994,BLKSIZE=27998,RECFM=VB)
```

Size Estimates for Exported EMRs

Generally, an exported EMR that contains all elements (not including JCL or documentation) uses approximately 30,000 bytes, or approximately one half of a track. This table provides some additional estimates for various export options:

Elements Exported	PageApprox. Size in Bytes	PageRecords Per 3390 Track
All elements except JCL/documentation are included in the export records; this includes OCCURS Hit data.	~ 30,000	~ 1.9
All elements except OCCURS Hit data and JCL/documentation are included in the export records (by default, OCCURS Hit data are excluded).	~ 2,800	~ 20
All elements except OCCURS Hit data, Accounting data, and JCL/documentation are included in the export records.	~ 2,545	~ 22
All elements except OCCURS Hit data, Accounting data, Accounting run list data, and JCL/documentation are included in the export records.	~ 2,153	~ 26

Note:

Size estimates do not include JCL or documentation—allow additional space for these types of data.

Size Estimates for Exported Calendar or Variable Records

Each exported calendar (CAL) or variable (VAR) record uses approximately 500 bytes, or approximately one-half a track for every 50 to 60 records.

Releasing Unused Space

When you allocate an export dataset, you can use the SPACE option with the RLSE parameter to force any unused space to be returned to the operating system at the end of the job step. For example:

```
SPACE=(TRK,(10,10),RLSE)
```

This example would allocate enough space for approximately 320 EMRs (not including JCL/documentation) or 17,920 VARs (assuming all secondary extents are allocated and filled with data).

Temporary and Permanent Datasets

You can allocate a temporary dataset instead of a permanent one. For example:

```
//WORK     DD DISP=(NEW,PASS),DSN=&&WORK,
//             DCB=(LRECL=27994,BLKSIZE=27998,RECFM=VB),
//             SPACE=(CYL,(50,50),RLSE),UNIT=SYSDA
```

You might prefer to do this, for example, when you need to make mass changes using an EXPORT step immediately followed by an IMPORT step that uses the exported data.

First, you export the records to the DD name WORK:

```
EXPORT EMR DD=WORK
```

Then, you can import the exported records back into the database either as updated or new records:

```
IMPORT EMR UPDATE DD=WORK
```

If you need to keep exported records for use in another operation at a later time, you can allocate permanent datasets. Be sure to allocate enough space for the records to be retained. For example:

```
//ALLOC   EXEC PGM=IEFBR14
//DD1      DD  DISP=(NEW,CATLG),
//         DSN=hlq.export.data,
//         DCB=(LRECL=27994,BLKSIZE=27998,RECFM=VB),
//         UNIT=SYSDA,SPACE=(CYL,(50,50))
```

In the import/export job, instead of using DSN=&&WORK (as for a temporary dataset), you use DSN=*hlq*.export.data to indicate a permanent dataset. Additionally, use DISP=OLD or DISP=SHR instead of DISP=NEW.

These sample statements in an import/export job specify the use of a temporary dataset:

```
//WORK     DD DISP=NEW,DSN=&&WORK,
//         DCB=(LRECL=27994,BLKSIZE=27998,RECFM=VB),
//         UNIT=SYSDA,SPACE=(CYL,(50,50))
```

This sample statement in an import/export job specifies the use of a permanent dataset:

```
//WORK     DD DISP=OLD,DSN=hlq.export.data
```

Install RI Server

To use the ZEKEXUTL utility program, you first must have RI Server installed, and the RI Server library must be included in the import/export JCL. Although RI Server is primarily used as a registration service, the libraries are also used for parsing incoming and outgoing XML during the import and export processes. RI Server must be installed on the system, but does not have to be running to use ZEKEXUTL. See the *ASG-RI Server Installation Guide* for installation instructions.

Configure the Zeke Server

The Zeke server enables ZEKEXUTL to import and/or export database records and must be installed and configured (i.e., through an environment dataset allocated to the Zeke address space). See the *ASG-Zeke Scheduling for z/OS Installation Guide* for detailed instructions.

Define OMVS Segment in User Profile

The ZEKEXUTL utility program uses Unix System Services (USS). It assumes the identity of its clients (i.e., acts as a surrogate of the z/OS user ID for users of Zeke when processing requests). For this module to assume the z/OS identity of a user that is located on z/OS, the client z/OS user ID must have a RACF OMVS segment defined in the profile for the user ID under which the Zeke started task runs. This is required for the module to access TCP/IP and to locate the Hierarchical File System (HFS) resources for TCP/IP (e.g., `'/etc/hosts'`).

Note:

If you are using another security product, an equivalent action is required.

As an administration aid, RACF provides the ability to define a default OMVS segment that can be associated with RACF user and group profiles. Refer to your *ASG-Zeke Scheduling for z/OS Installation Guide* for more information.

Executing the ZEKEXUTL Utility

The Zeke database is allocated to the Zeke started task and subsystem name. Because the ZEKEXUTL utility program communicates with Zeke via its subsystem name, Zeke must be active for the program to be executed.

To execute the ZEKEXUTL utility

- Create a jobstream that executes the ZEKEXUTL procedure and include control statements that specify the appropriate import and/or export commands (see Figure 2 on page 121).

  See “Control Statements” on page 122 for rules and guidelines for entering control statements to be used for import/export functions.

  For EXPORT command syntax and parameters, see “EXPORT—Export Database Records to a File” on page 127. For IMPORT command syntax and parameters, see “IMPORT—Import Records from a File or Database” on page 134.

Figure 2 illustrates the general JCL requirements:

Figure 2 • ZEKEXUTL General JCL Requirements

```
        //your job card here
        //*
(1)     //ZEKEXUTL EXEC PGM=ZEKEXUTL,REGION=0M,
(2)     //  PARM='SUBSYS=xxxx,DATASPACE'
(3)     //STEPLIB  DD DISP=SHR,DSN=your.zeke.LINKLIB
        //         DD DISP=SHR,DSN=your.oasis.LINKLIB
        //         DD DISP=SHR,DSN=your.ris.LINKLIB
        //*
(4)     //SYSPRINT DD SYSOUT=*
(5)     //SYSOUT   DD SYSOUT=*
(6)     //SYSIN DD *
        import/export control statements
        $$
        /*
        //*
(7)     //ENVIRON DD *
        TZ=EST5EDT
        /*
        //
```

This table describes each of the statements highlighted in Figure 2 on page 121:

Statement	Description
(1)	The import/export utility module is ZEKEXUTL. The utility uses storage above and below the 16 MB line. Execute the statement with REGION=0M.
(2)	The PARM statement must include your Zeke subsystem. The Zeke database is allocated to that subsystem and is not part of the import/export utility job control. If you have a large number of records to process, using a data space improves performance. Include the DATASPACE keyword to have the utility request a data space for database I/O.
(3)	The STEBLIB concatenates the Zeke, OASIS, and RI Server load libraries. These libraries contain modules (DLLs) that are used by the import/export utility.
(4)	Typically, the SYSPRINT DD name is allocated to SYSOUT. It will receive error messages and the report output (unless the report is directed to another DD name).
(5)	The SYSOUT DD name is the log and receives trace information.
(6)	The SYSIN DD name contains the control statements for the import/export utility.

Statement	Description
(7)	ASG recommends you include the ENVIRON DD name if your location's time zone is other than GMT. This should contain a setting for the TZ environment variable. (See your IBM/LE documentation for instructions on how to set the TZ environment variable for your location.)

LE Runtime Options

By default, Language Environment (LE) runtime options are set for a typical LE environment. If necessary, you can override the runtime options for the Import/Export program to suit your environment. See Appendix B, "LE Runtime Options," on page 623 for details.

Caution! ASG recommends that you do not override the LE program runtime options unless you have extensive LE experience/knowledge. Changing the LE options could have unpredictable/harmful results and can cause the program to become unstable.

Log Messages

To prevent excessive log/trace output (which could slow processing), log messages are written to the log only when enabled by the ZEKE_TRACE environment variable. This variable controls the types of trace messages the Zeke server logs to the trace log. For production systems, ASG recommends setting the ZEKE_TRACE variable to `/all` (i.e., no trace messages are logged). Refer to your *ASG-Zeke Scheduling for z/OS Installation Guide* for information about the ZEKE_TRACE variable.

Control Statements

Import/export control statements are presented to the SYSIN for the import/export utility as free form text.

General Format

Consider these formatting guidelines for your control statements:

- You can include as many leading space characters as desired to improve readability.You can continue control statements from line to line using plus (+) or dash (-) following the last word of a control statement. The continuation character must be preceded by at least one space and be the last symbol on the line to be continued to the next line. For example:

```
IF (ENAME="TEST*" AND SYS="DEV" AND USERID="DV123") + THEN ACCEPT
```

- You can enter control statements and keywords in upper, lower, or mixed case; however, the values are case-sensitive. For example, these two statements are equivalent:

```
if (ename=TEST* and sys=DEV and userid='DV123') then accept
If (Ename="TEST*" AnD Sys="DEV" aNd userID="DV123") + Then Accept
```

- If the value contains spaces or punctuation characters, or starts with a numeric character, then you must surround the value in either single quotes (`' '`) or double quotes (`" "`). Otherwise, quotes are not required. If you are unsure whether values to be changed might contain these characters, ASG recommends that you use either single or double quotes.

  For example, these statements are equivalent (where the asterisk [`*`] is considered a wildcard character):

```
"TEST*"
'TEST*'
TEST*
```

  These statements are *not* equivalent:

```
"John Doe"
John Doe

"Doe,John"
Doe,John
"2DEV"
2DEV
```

- You can construct a compound IF statement by including `AND` or `OR` between the expressions in the IF statement. For example:

```
IF (ENAME="TEST*" AND SYS="DEV") THEN ACCEPT
```

- Use parentheses to group compound expressions together, so that they will be evaluated in the correct context. For example:

```
IF ((ENAME="TEST*" AND SYS="DEV") OR SYS="DV12")) THEN ACCEPT
```

- A statement that begins with an asterisk (*) or a pound sign (#) is a comment and is not executed by the utility. You can also include comments anywhere on a line using /* or // as the starting delimiter. For example:

```
CHANGE ONLY /*The rest of this line is a comment.*/
CHANGE ONLY //The rest of this line is a comment.
*This line is a comment.
#This line is a comment.
```

Wildcard Values

Values can include wildcard substitution characters when used as part of the IF expression to be evaluated. An asterisk (*) will match zero or more characters of any type. A question mark (?) will match any single character value.

Examples:

This matches ABC or ABCD or ABC 123:

```
"ABC*"
```

This matches ABCD, but not ABC or ABC 123:

```
"ABC?"
```

This matches ABCD or ABC 123, but not ABC:

```
"ABC?*"
```

This matches ABC 123, but not ABC or ABCD:

```
"*123"
```

This matches ABC 123, but not ABC or ABCD:

```
"* *"
```

This matches ABC 123, but not ABC or ABCD:

```
"A*2?"
```

This matches ABCD, but not ABC or ABC 123:

```
"????"
```

This matches anything (including an empty value):

```
"*"
```

Required Statements

Either the IMPORT or EXPORT control statement is required; all other control statements are optional.

Do not enter both the IMPORT and EXPORT statements in the same group of control statements.

When you enter an IMPORT or EXPORT statement that refers to a DD name, you also must include the DD name as part of the JCL for the import/export utility. For example:

```
...
//MYDATA   DD DISP=OLD,DSN=MY.EXPORT.DATA
//SYSIN    DD *
IMPORT EMR ADD DD=MYDATA
...
```

Logical Steps

You can include multiple logical steps in a single execution using the $$ control statement to indicate the end of a set of control statements. Processing for statements up to the $$ control statement begins as soon as the $$ statement is reached. After each set of statements is processed and completed successfully, the next set is processed.

For example:

```
...
//WORK     DD DISP=(NEW,PASS),DSN=&&WORK,
//            DCB=(LRECL=27994,BLKSIZE=27998,RECFM=VB),
//            SPACE=(CYL,(50,50),RLSE),UNIT=SYSDA
//SYSIN    DD *
EXPORT EMR DD=WORK
  FILTER REJECT
    IF (ENAME="PAY*" AND +
        SYS  ="TEST" AND +
        USER ="USER1") THEN ACCEPT
$$
IMPORT EMR UPDATE DD=WORK
  CHANGE ONLY
    IF (JOB) THEN ENAME="SAMPLE", SYS="PRD2"
$$
/*
...
```

XPath Fields

All field names within EMR, CAL, and VAR records are represented internally as XML elements and attributes. These elements and attributes can be located using XML paths or *XPath* names. Using XPath names allows you to reference any field within a record. Any field within the XML record can be referenced by its XPath name and can be used in control statements in place of the common field name.

For example, this statement:

```
IF (ENAME="TEST*" AND SYS="DEV" AND USERID="DV123") THEN ACCEPT
```

can be represented as:

```
IF ("descendant::emrBaseData/@eventName"   = "TEST*" AND +
    "descendant::emrBaseData/@sysID"       = "DEV"   AND +
    "descendant::emrBaseData/@userID"      = "DV123" )   +
        THEN ACCEPT
```

Note:

When using a XPath name as a field name, you must enclose the XPath name in single or double quotes.

When you supply a field name as part of a control statement, the field name is converted internally to its equivalent XPath name. The XPath field names provide a way to navigate through the elements and attributes that make up EMR, CAL, and VAR records.

See Appendix A, “XPath Location Paths,” on page 609 for more information.

EXPORT—Export Database Records to a File

This function exports records from the Zeke database to an external file as XML records. You can export these types of records:

- Event master records (EMR)
- Calendar records (CAL)
- Variable records (VAR)

Exported records are written to the specified DD name or dataset name as XML elements and attributes. The entire set of exported records represent an XML document. After the records have been written to a DD name or dataset, you can use an external XML utility to modify or use the XML document (as needed for other processing requirements beyond Zeke database import/export functions).

Note:

If there is insufficient LRECL for the output dataset when the utility attempts to write a record to the XML file, the utility issues an error message and terminates.

You can use the EXPORT function to perform these actions:

- Migrate data records from one database to another (typically, from a test database to a production database). Both databases can be at the same or different Zeke release levels. For different releases, any data attributes that are not common to both releases are ignored.
- Make mass changes against existing records. You can change values in one record and apply the changes across a broad range of records. You use the EXPORT function to specify the data to be changed and export the selected records to a file. You can select whether to export all fields or only the changed fields plus the required fields. Then, you can use the IMPORT function to apply the changes to the original records in the database.
- Export records from the Zeke database as backups that can be restored later. If needed later, you can use the IMPORT function to import one or more of these backup records back into the original database. When you do so, the record in the Zeke database is replaced with the imported record. If changes were made to the record in the database after the record was exported, those changes are overwritten.

Caution! Not all fields can be imported back into Zeke. You cannot import or update fields that are set internally by Zeke (e.g., OCCURS Hit, event accounting information for EMRs, and variable accounting information for VARs). Records that contain these values are still imported, but these system-set field values are not processed.

Note:

See "IMPORT—Import Records from a File or Database" on page 134 for details about importing records.

When exporting records, the event number is the key identifier for each record and always is exported. When importing records back into their original database, the event number is used to find and update the original record. When importing records to another database, the original event number is not re-used; a new event number is created on the new database.

EMR Export Considerations

Before you use the EXPORT utility to export EMRs, ASG recommends that, first, you run a LIST EVENTS report (using the same specifications you plan to use for the export) and review the event information currently in the Zeke database. (See "LIST EVENTS/PLAN—EMR/SQR Listings" on page 213.) Check the names and values you plan to export for modification and ensure that they are specified accurately when used with the EXPORT utility. Field values must match exactly.

Consider how these EMR attributes are handled during the export process:

- EMRs that include the REFEVENT keyword in the OCCURS clause are exported normally; however, you have the option of including or excluding these EMRs for an import. If you choose to include them, be aware that the import EMRs could now refer to events that no longer exist or that have been updated. It is likely that these EMRs will require manual updates via the Zeke online facility (or OpsCentral) to correct the REFEVENTs in the OCCURS clauses.
- For EMRs with non-Zeke JCL to be exported completely, the EXPORT job must include the appropriate DD names for the JCL sources (in the same way as in the Zeke started task). If the JCL sources are omitted from the EXPORT job, these messages will appear in the EXPORT job log for every event with a non-Zeke JCL source:

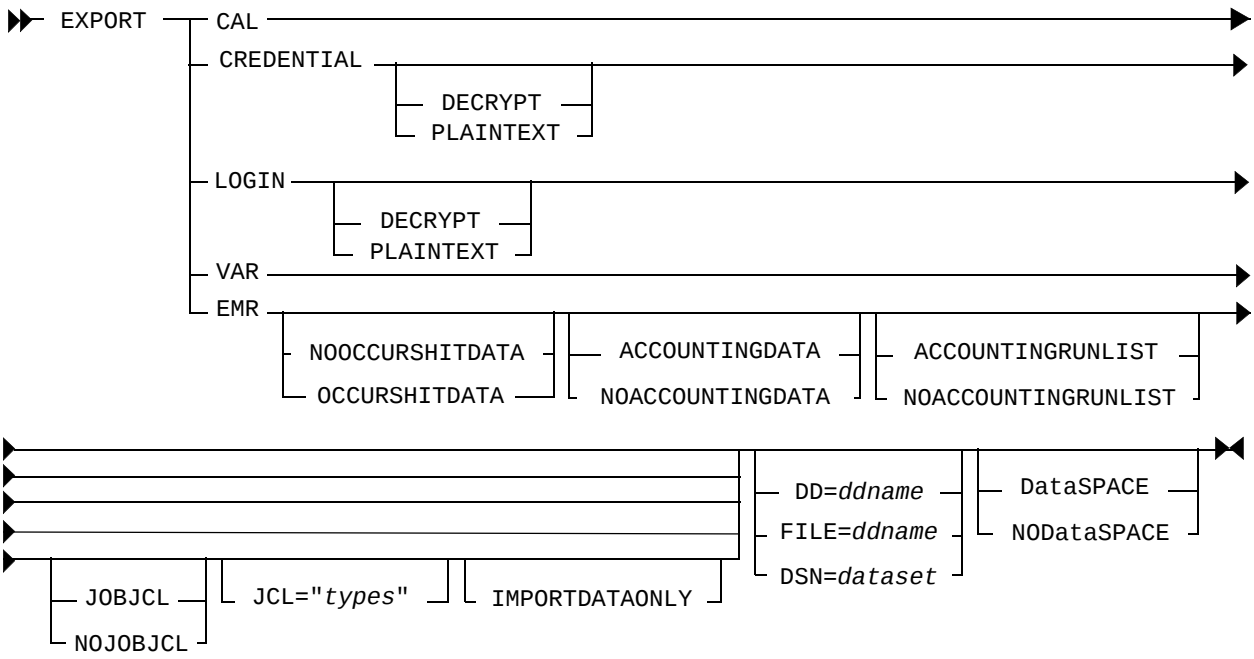
  ```
  Z0658E Event 000001 ver 00000 ZEKE06N error RC=810
  Z4658E Event 000004 ver 00000 load failed, module name is ZEKE06O
  ```

  This message will appear (for each event) in the log (SYSOUT DD name) for the EXPORT job:

  ```
  Z6057W JCL for event 1 is unavailable
  ```

  The EMRs that are missing the JCL source specification still are exported, but without their JCL.

Syntax



See "Control Statements" on page 122 for more information (including general formatting rules and guidelines).

Parameters

These are the valid parameters for the EXPORT function:

Parameter	Description	
CAL	Indicates to export calendar records only.	
CREDential	Indicates to export credential records only.	
	DECRYPT PLAINTEXT	Indicates to decrypt sensitive values (i.e. passwords). If this keyword is omitted, such values are encrypted.
LOGIN	Indicates to export login records only. These are the optional keywords:	
	DECRYPT PLAINTEXT	Indicates to decrypt sensitive values (i.e. passwords). If this keyword is omitted, such values are encrypted.
VAR	Indicates to export variable records only.	

Parameter	Description
EMR	Indicates to export event master records (EMRs) only. These optional keywords can be used to specify the types of data to include/exclude in the export file: **NOOCCURSHITDATA** — Default. Indicates to exclude the OCCURS Hit data from the exported EMR. **OCCURSHITDATA** — Indicates to include the OCCURS Hit data in the exported EMR. **Caution!** Because using the OCCURSHITDATA parameter could greatly increase export time, use this parameter only when necessary. **ACCOUNTINGDATA** — Default. Indicates to include event accounting data in the exported EMR. For example: • Number of times the event has been dispatched. • Last date/time the EMR was updated. • User who updated it last. • Start and end dates/times of the last execution that ran to completion (unless the job has never been dispatched). • Status of the last execution that ran to completion. **NOACCOUNTINGDATA** — Indicates to exclude event accounting data from the exported EMR. **ACCOUNTINGRUNLIST** — Default. Indicates to include accounting run list data in the exported EMR. For example: • Dispatch data and time for the last three dispatches of the event. • Tape drive and virtual memory usage. • Duration. • Amount of CPU time. • Completion code. **NOACCOUNTINGRUNLIST** — Indicates to exclude accounting run list data from the exported EMR.

Parameter	Description	
	JOBJCL	Default. For job events, indicates to include JCL in the exported EMR. To include non-Zeke JCL, specify the JCL types using the JCL= keyword.
	NOJOBJCL	Indicates to exclude JCL from the exported EMR.
	JCL="*types*"	Specifies the types of JCL to include in the exported EMR for job events. See "EMR Export Considerations" on page 128 before using this option. The default value is "ZEKE". **Note:** This parameter is ignored if the NOJOBJCL parameter also has been specified. These are the valid JCL types. Multiple types can be specified (separated by commas). ZEKE Default. Zeke JCL. **Caution!** Non-Zeke JCL cannot be imported into a Zeke database. ALL All JCL types. BIM Bim-Edit JCL. CONDR Condor JCL. DRV CA-Driver. ICCF ICCF JCL. LIBR Librarian JCL. OWL OWL JCL. PAN Panvalet JCL. PDS JCL from partitioned dataset members.

Parameter	Description		
		`SLB`	SLIBOOK JCL.
		`SLI`	SLI JCL.
		`VOL`	Vollie JCL.
		`X14C`	JCL supplied by the ZEKE14C user exit from a Zeke-unsupported JCL source.
	`IMPORTDATAONLY`		Indicates that the exported EMR will contain only data that can be imported into Zeke using the IMPORT EMR command. With this option, any other optional keywords for the EMR parameter are ignored. Specifying IMPORTDATAONLY is equivalent to specifying these keywords: `JOBJCL JCL="ZEKE" NOOCCURSHITDATA NOACCOUNTINGDATA NOACCOUNTINGRUNLIST`
`DD=`*`ddname`* `FILE=`*`ddname`*	Either of these parameters specifies the DD name of the dataset to which to send the exported records.		
`DSN=`*`dataset`*	Specifies the name of the dataset to which to send the exported records.		
Note: If you do not specify a file or dataset option, the output is written to the SYSOUT DD.			
`DataSPACE`	Recommended. Indicates to switch the mode for accessing the Zeke database, so that the database is reopened in a data space. (Database access via the data space increases the speed of I/O.) After a data space is created, database access continues via the data space until the data space is closed by a NODSPACE option or at end-of-program. **Note:** If a data space cannot be created for the Zeke database (e.g., if your system is not configured to permit data space creation or in the case of insufficient memory or resources), the utility will attempt to open the Zeke database using normal I/O.		

Parameter	Description
NODataSPACE	Indicates to switch the mode for accessing the Zeke database, so that a data space is not used. If a data space previously existed, access returns to normal I/O and the data space is destroyed. After a data space is terminated, database access continues to use normal I/O until a data space is created by a DSPACE option or at end-of-program.

EXPORT Examples

```
EXPORT EMR FILE=XMLOUT
REPORT DD=XMLRPT
FILTER REJECT
IF (ENAME=TEST*) THEN REJECT
IF (SYSTEM=ABC OR SYSTEM=XYZ) THEN ACCEPT
CHANGE ONLY
IF (SYSTEM=ABC AND ENAME=PAY*) THEN +
    (APPL=PAYROLL PRIORITY=5)
IF (SYSTEM=ABC OR ENAME=PAY*) THEN +
   (SYSTEM=XYZ PRIORITY=2)
$$
```

```
EXPORT EMR FILE=XMLOUT
FILTER REJECT
IF (SYSTEM=ABC OR SYSTEM=XYZ) THEN ACCEPT
IF (ENAME=TST*) THEN REJECT
CHANGE ONLY
IF (SYSTEM=ABC AND ENAME=PAY*) THEN (APPL=PAYROLL PRIORITY=5)
IF (SYSTEM=ABC OR ENAME=PAY*) THEN (SYSTEM=XYZ PRIORITY=2)
$$
```

See “Sample Import and Export Jobs” on page 160 for more examples.

IMPORT—Import Records from a File or Database

The IMPORT function reads XML records that were exported previously to a dataset and imports the records to the Zeke database. Depending on the options specified, imported records can be added to the Zeke database or can be used to update existing records.

Caution! Not all fields can be imported. You cannot import or update fields that are set internally by Zeke (e.g., OCCURS Hit, event accounting information, and variable accounting information). Records that contain these values still are imported, but these system-set field values are not processed.

You can use the IMPORT function to perform these actions:

- Import data records exported from one Zeke database into another (typically, from a test database to a production database). Both databases can be at the same or different Zeke release levels. For different releases, any data attributes that are not common to both releases are ignored.
- Make mass changes against existing records. You can change one or more values in a record and apply that change across a broad range of records. After exporting the records and making the desired changes, you use the IMPORT function to apply the changes to the original records in the Zeke database.
- Restore previously exported backup records to the Zeke database. When you do so, the record in the database is replaced with the imported record. If changes were made to the record in the database after the record was exported (i.e., before it was backed up), those changes are overwritten.

Note:

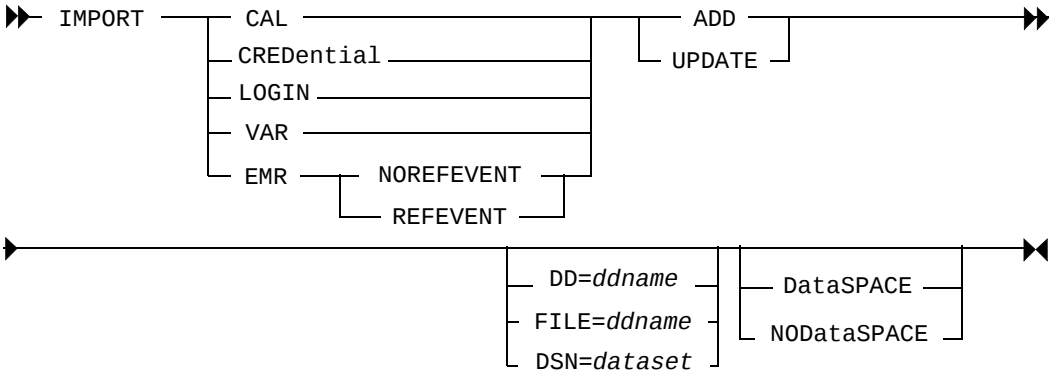
See "EXPORT—Export Database Records to a File" on page 127 for details about exporting records.

When importing records back into their original database, the event number is the key identifier for each record and is used to find and update the original record. When importing records to another database, the original event number is not re-used; a new event number is created on the new database.

EMR Import Considerations

Before you use the IMPORT utility to import events, ASG recommends that, first, you run a LIST EVENTS report (using the same specifications you plan to use for the export) and review the event information currently in the Zeke database. (See "LIST EVENTS/PLAN—EMR/SQR Listings" on page 213.) Check the names and values you plan to export for modification and ensure that they are specified accurately when used with the EXPORT utility. Field values must match exactly.

Syntax



See "Control Statements" on page 122 for more information (including general formatting rules and guidelines).

Parameters

These are the valid parameters for the IMPORT function:

Parameter	Description
CAL	Indicates to import calendar records only.
CREDential	Indicates to import credential records only.
LOGIN	Indicates to import login records only.
VAR	Indicates to import variable records only.
EMR	Indicates to import EMRs only.
NOREFEVENT	Default. Valid only with the EMR parameter. This option rejects any EMRs that include the REFEVENT keyword in the OCCURS clause. If such a record is encountered during import, a message is issued to the log file (and to the report file, depending on the REPORT options) indicating the event number of the record that was rejected. **Note:** If an OCCURS clause contains a REFEVENT reference to an event name, no warning is issued.

Parameter	Description
REFEVENT	Valid only with the EMR parameter. This option enables EMRs that include the REFEVENT keyword in the OCCURS clause to be imported. Any time one of these records is encountered during import, a message is issued to the log file (and to the report file, depending on the REPORT options) indicating the event number or event name of the record that was imported and warning that event references might no longer be accurate. **Caution!** If an OCCURS clause for an imported event refers to another event, the resulting imported EMR could refer to an event that does not exist or has changed number. The imported EMR might require manual verification and updating to correct its OCCURS (REFEVENT) clause.
ADD	Indicates to add the imported records to the Zeke database.
UPDATE	Indicates to update the existing records in the Zeke database with the imported records.
DD=*ddname* FILE=*ddname*	Either of these parameters specifies the DD name of the dataset that contains the records to be imported.
DSN=*dataset*	Specifies the name of the dataset that contains the records to be imported.
DataSPACE	Recommended. Indicates to switch the mode for accessing the Zeke database, so that the database is reopened in a data space. Database access via the data space increases the speed of I/O. After a data space is created, database access continues via the data space until the data space is closed by a NODSPACE option or at end-of-program. **Note:** If a data space cannot be created for the Zeke database (e.g., if your system is not configured to permit data space creation or in the case of insufficient memory or resources), the utility will attempt to open the Zeke database using the normal I/O.
NODataSPACE	Indicates to switch the mode for accessing the Zeke database, so that a data space is not used. If a data space previously existed, access returns to normal I/O and the data space is destroyed. After a data space is terminated, database access continues to use normal I/O until a data space is created by a DSPACE option or at end-of-program.

IMPORT Example

```
IMPORT EMR ADD FILE=XMLOUT
FILTER REJECT
IF (ENAME=TEST*) THEN REJECT
IF (SYSTEM=ABC OR SYSTEM=XYZ) THEN ACCEPT
CHANGE ONLY
IF (SYSTEM=ABC AND ENAME=PAY*) THEN +
    (APPL=PAYROLL PRIORITY=5)
IF (SYSTEM=ABC OR ENAME=PAY*) THEN +
   (SYSTEM=XYZ PRIORITY=2)
$$
```

See “Sample Import and Export Jobs” on page 160 for more examples.

FILTER—Filter Imported/Exported Records

The FILTER commands enables you to select records for export or import based on filtering rules. You use IF/THEN statements to establish these rules.

When used with the EXPORT control statement, records are selected for exporting from the Zeke database to an external dataset as XML elements and attributes.

When used with the IMPORT control statement, records are selected for importing from an external file into the Zeke database.

If no filtering control statement is included for an import/export, then all available records are selected by default.

You can filter based on these record data:

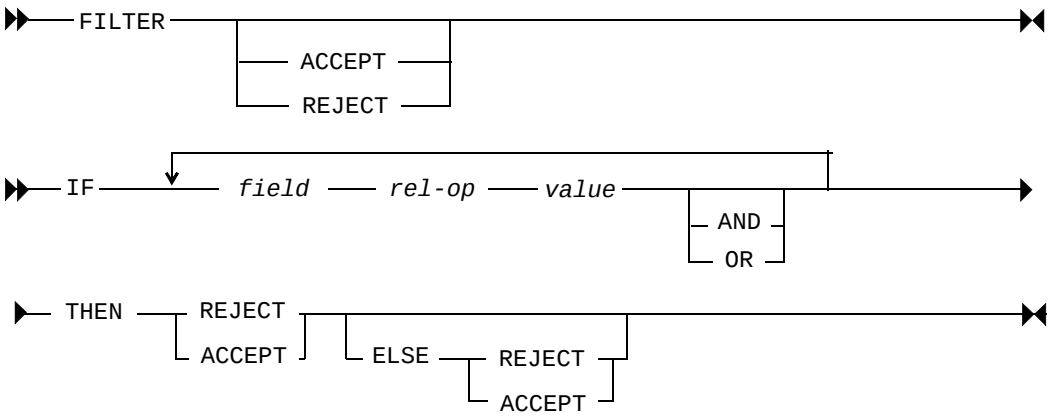
- EMR fields

  Note:

  If a field included in a filtering rule is not valid for the event (e.g., because of its event type), the record is considered to be a mismatch for that filtering rule.

- A string in an OCCURS clause
- A string in a WHEN condition
- Job data
- Resources
- A JCL string
- Any field in the JCL source
- Calendar fields
- Variable name or type

Syntax



See "Control Statements" on page 122 for more information (including general formatting rules and guidelines).

Parameters

These are the valid parameters for the FILTER function:

Parameter	Description
ACCEPT	When used with the FILTER function, this keyword indicates to accept all records that are not rejected by the filtering rule in the IF/THEN statement. In an IF/THEN statement, this keyword indicates to accept all records that match the filtering rule.
REJECT	Default. When used with the FILTER function, this keyword indicates to discard all records that are not accepted by the filtering rule in the IF/THEN statement. If neither the ACCEPT nor the REJECT parameter is included, then REJECT is assumed. In an IF/THEN statement, this keyword indicates to discard all records that match the filtering rules.
IF	Precedes the selection criteria for the filtering rule. The IF statement indicates to select records with the specified field values and must be followed by a THEN statement. **Note:** When using AND or OR to form compound IF statements, be sure to use parentheses to group the expressions together so that they will be evaluated in the correct context.

Parameter	Description
THEN	Precedes the action to take (i.e., ACCEPT or REJECT) for the selected records and must be preceded by an IF statement.
ELSE	Optional. Precedes the action to take (i.e., ACCEPT or REJECT) for the records *not* selected by the IF statement and must be preceded by an IF/THEN statement.
field	Specifies the field to match (i.e., any LIST EVENT or LIST VARIABLE parameter or an XPath location path). For best performance, ASG recommends that you specify event fields from the EMR directory (EDB), as applicable, in your export filtering criteria. Using these fields for filtering significantly reduces database I/O because a complete EMR is not required. These are the common selection fields: ACTive APPL DEACtivate ENAMe EVENT EVTCal GROupid JOB JOBN SYStem USERid
rel-op	Relational operator. These are the valid values: EQ Equal to = == NE Not equal to != ^= GT Greater than > GE Greater than or equal to >= LT Less than < LE Less than or equal to <=

Parameter	Description
value	Specifies the field value (i.e., numeric value, word, or string) to match. **Note:** In numerical comparisons, a blank or empty field has a value of zero.

Filtering Rules

This section explains filter rule processing.

Evaluation of Filtering Rules

The FILTER ACCEPT or FILTER REJECT command is applied only after all of the filtering rules have been evaluated and is applied only to records that are not accounted for in the rules. If the filtering rules provide a conclusive outcome (i.e., accept or reject) for every record, then the FILTER ACCEPT or FILTER REJECT command is ignored. (Rules that contain an ELSE statement always provide a conclusive outcome for all records.) If some records do not match any of the filtering rules, and there is no ELSE statement, the FILTER ACCEPT or FILTER REJECT command is used to determine whether those records will be processed or discarded.

Examples:

This accepts only the records with job names that do not begin with DEV or TEST:

```
FILTER REJECT
IF (JOBNAME != 'DEV*' AND JOBNAME NE 'TEST*') THEN ACCEPT
```

This accepts only the records for event numbers 50 through 122:

```
FILTER REJECT
IF (EVENT GE 50 AND EVENT LE 122) THEN ACCEPT
```

Either of these statements accepts all records except for event numbers 50 through 122:

```
FILTER ACCEPT
IF (EVENT >= 50 AND EVENT <= 122) THEN REJECT
```

Or

```
FILTER REJECT
IF (EVENT < 50 OR EVENT > 122) THEN ACCEPT
```

This accepts all records (any records not accepted by the IF statement are accepted by the FILTER ACCEPT; therefore, the IF statement is redundant):

```
FILTER ACCEPT
IF (APPL='XXXX') THEN ACCEPT
```

This accepts only the records with an application ID of XXXX (the ELSE REJECT overrides the FILTER ACCEPT):

```
FILTER ACCEPT
IF (APPL='XXXX') THEN ACCEPT ELSE REJECT
```

Using Multiple Filter Rules

If you need to specify multiple filtering rules in a single FILTER operation, consider these guidelines for how rules are processed:

- If a record matches more than one filtering rule, the first matching rule is used to determine the outcome for that record. Any subsequent filtering rules that match this record are ignored.
- If you have three filtering rules, and all records are included in either the first or second rule, then the third rule is not evaluated at all. A rule that contains an ELSE statement always will provide an outcome for every record, so any subsequent filtering rules will always be ignored.

Example 1:

```
FILTER ACCEPT
IF (event='1234') THEN ACCEPT ELSE REJECT
IF (event='5678') THEN REJECT ELSE ACCEPT
```

The first rule contains an ELSE statement, and therefore provides an outcome for every record, based on whether or not its event number is `1234`. The second rule and the FILTER ACCEPT command both are ignored.

Example 2:

```
FILTER REJECT
IF (event='1234') THEN ACCEPT
IF (event='5678') THEN REJECT ELSE ACCEPT
```

The second rule is evaluated for any records whose event number is not `1234`. Any record with an event number of `5678` is rejected; all others are accepted. The FILTER REJECT command is ignored. (The same results can be accomplished by omitting the first IF rule.)

Example 3:

```
FILTER ACCEPT
IF (event='1234') THEN REJECT
IF (sys='PROD') THEN REJECT
```

The two filtering rules will cause all records with either an event number of `1234` or a system ID of `PROD` to be rejected. The FILTER ACCEPT command is applied to all other records.

Example 4:

```
FILTER REJECT
IF (event='1234') THEN ACCEPT
IF (event='5678') THEN REJECT
```

Only records with an event number of `1234` are accepted; all others are rejected. The FILTER REJECT command is applied to all records whose event number is not `1234` or `5678`. (The same results can be accomplished by omitting the second IF rule.)

Example 5:

```
FILTER REJECT
IF (event='1234') THEN ACCEPT
IF (sys='PROD') THEN REJECT ELSE ACCEPT
```

Records with an event number of `1234` are accepted. Of the remaining records, those which have a system ID of `PROD` are rejected. All other records are accepted. (A record with *both* `event=1234` *and* `sys=PROD` is accepted.) The FILTER REJECT command is ignored.

Example 6:

```
FILTER REJECT
IF (event='1234') THEN ACCEPT
IF (event='5678') THEN ACCEPT ELSE REJECT
```

All records with an event number of `1234` or `5678` are accepted; all others are rejected by the ELSE statement. The FILTER REJECT command is ignored. (The same results can be accomplished by omitting the ELSE statement, in which case the FILTER REJECT command would be applied to any remaining records.)

CHANGE—Change Imported Record Values

The CHANGE command enables you to change field values in records being processed by the import/export utility. The CHANGE command is used to indicate whether to import/export *all* data fields for a record, or only the required fields and changed fields. If you want to change fields for specific records (rather than *all* records), you can use IF/THEN statements to establish *change rules*.

You can change record data such as these:

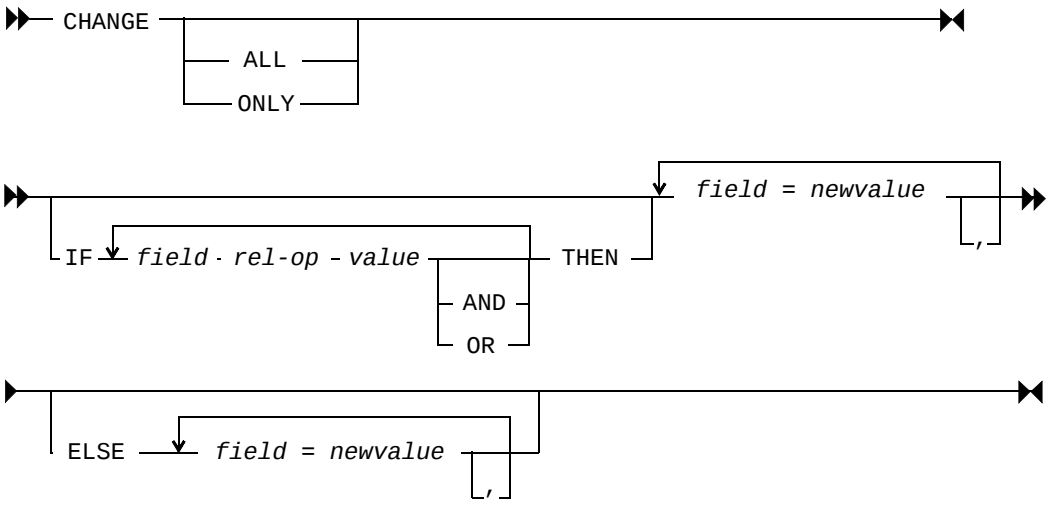
- EMR fields

  Note:

  If a field included in a change rule is not valid for the event (e.g., because of its event type), the record is considered to be a mismatch for that change rule.

- Job data
- Resources
- Calendar record fields
- Variable record fields

Syntax



Note:

When using AND or OR to form compound IF statements, be sure to use parentheses to group the expressions together so that they will be evaluated in the correct context.

If you are not using an IF/THEN statement (because you want to change *all* records), do *not* enclose the field and new value in parentheses.

See "Control Statements" on page 122 for more information (including general formatting rules and guidelines).

Caution! When using the import/export utility to change XML field values, be aware that IF statements operate at the record level, not at the field level. Therefore, if a record contains multiple fields with the same name as the field being changed, *all* those fields will be changed.

Consider this example:

```
EXPORT EMR DD=EMR IMPORTDATAONLY
FILTER REJECT
IF (".//JCLFIELDDATA/@LABEL" = "DD NAME:" AND +
    ".//JCLFIELDDATA/@VALUE" = "$JCL3") +
  THEN ACCEPT
CHANGE ALL
IF (".//JCLFIELDDATA/@LABEL" = "DD NAME:") +
  THEN ".//JCLFIELDDATA/@VALUE" = "CHANGED"
$$
```

Let's assume this is the XML:

```
<jclList>
  <JCL sourcme="CMS" readOnly="1" numFields="2" isActive="0">
    <JCLFieldData label="Fname:" length="8" value=""/>
    <JCLFieldData label="Ftype:" length="8" value=""/>
  </JCL>
  <JCL sourcme="Zeke JCL" readOnly="0" numFields="0" isActive="0"/>
  <JCL sourcme="BIM Edit" readOnly="1" numFields="2" isActive="0">
    <JCLFieldData label="Library:" length="16" value=""/>
    <JCLFieldData label="Member:" length="16" value=""/>
  </JCL>
  <JCL sourcme="PDS" readOnly="1" numFields="2" isActive="1">
    <JCLFieldData label="DD Name:" length="8" value="ZEKEJCL"/>
    <JCLFieldData label="Member:" length="8" value="GENERIC"/>
  </JCL>
</jclList>
```

You can use XPath predicates to control which elements and attributes will be changed. A predicate is like an IF statement condition expression embedded within the square brackets `[...predicate...]` for an element. All names in the predicate must be XPath element or attribute names and must be quoted. See Appendix A, "XPath Location Paths," on page 609 for a description of XPath predicates.

For example:

```
...
EXPORT EMR DD=EMR IMPORTDATAONLY
FILTER REJECT
IF (".//JCLFIELDDATA/@LABEL" = "DD NAME:" AND +
     ".//JCLFIELDDATA/@VALUE" = "$JCL3") +
    THEN ACCEPT
CHANGE ALL
    '//JCLFieldData["./@label" = "DD*" AND +
    "./@value"="$JCL3"]/@value' = "PRODJCL"
$$
```

In this example, only JCLFieldData elements with a label attribute matching `"DD*"` and a value attribute that is `"$JCL3"` will have their value attribute changed to `"PRODJCL"`.

You also can make changes to the exported XML using a text editor or an XSLT utility program using Extensible Style-sheet Language (ESL).

Parameters

These are the valid parameters for the CHANGE function:

Parameter	Description
ALL	Default. Indicates that the resulting record data will contain all fields.
ONLY	Indicates that the resulting record data will contain the required fields along with any fields changed by the change rules. If neither the ALL nor ONLY parameter is included, ALL is assumed.

Parameter	Description
`IF`	Precedes the selection criteria for the change rule. The IF statement indicates to select records with the specified field values and must be followed by a THEN statement. If you are updating all records, then an IF/THEN statement is not required. **Note:** When using AND or OR to form compound IF statements, be sure to use parentheses to group the expressions together so that they will be evaluated in the correct context.
`THEN`	Precedes the new field values for the selected records. A THEN statement must be preceded by an IF statement. If you are updating all records, then an IF/THEN statement is not required.
`ELSE`	Precedes the new field values for the records *not* selected by the IF statement and must be preceded by an IF/THEN statement.
`field`	In an IF statement, this specifies the field to match. In a THEN statement, this is the field to change. This value can be any LIST EVENT parameter or XPath location path. **Note:** If you are not using an IF/THEN statement (because you want to change *all* records), do *not* enclose the field and new value in parentheses. **Caution!** If a record contains multiple fields with the same name as the field being changed, ALL of those fields will be changed. See the caution on page 145 for more information.
`rel-op`	Specifies the relational operator. These are the valid values: `EQ` `=` `==` Equal to `NE` `!=` `^=` Not equal to `GT` `>` Greater than `GE` `>=` Greater than or equal to

Parameter	Description
	LT Less than LE Less than or equal to <=
value	In an IF statement, this specifies the field value (i.e., numeric value, word, or string) to match. **Note:** In numerical comparisons, a blank or empty field has a value of zero.
newvalue	In a THEN or ELSE statement, this specifies the new field value. If you are not using an IF/THEN statement (because you want to change *all* records), do *not* enclose the field and new value in parentheses.
,	Specifies the separator to use between field/value pairs.

Examples:

This changes the system ID to PROD only if the existing system ID *is not* TEST:

```
CHANGE ALL
IF (SYS NE 'TEST') THEN SYS EQ 'PROD'
```

This changes the system ID to PROD only if the event name ends in P:

```
CHANGE ALL
IF (ENAME == '*P') THEN SYS = 'PROD'
```

This changes the system ID to PROD for event names that begin with PAY or job names that do not begin with DEV or TEST:

```
CHANGE ONLY
IF (ENAME = 'PAY*' OR +
   (JOBNAME NE 'DEV*' AND JOBNAME ^='TEST*')) THEN +
    SYSTEM = 'PROD'
```

This changes the SYS and GROUP fields for *all* records. An IF/THEN statement is not required when changing *all* records. The SYS and GROUP parameters can also appear on one line, separated by a comma.

```
CHANGE ONLY
SYS='PROD'
GROUP='ABC'
```

Note:

Do not use parentheses in expressions that do not contain an IF/THEN statement.

Changes the group ID to ABC and the event name to DEF for records whose system ID is PROD. For all other records, the group ID is changed to GHI and the event name is changed to DEF.

```
CHANGE ONLY
IF (SYS='PROD') THEN GROUP='ABC',ENAME='DEF' ELSE +
   GROUP='GHI',ENAME='JKL'
```

CHANGE Examples

These examples illustrate changing the JCL source for EMRs.

This example illustrates exporting and then changing JCL source for job events that have JCL stored in a PDS with ddname TESTJCL to now use ddname PRODJCL.

```
...
EXPORT EMR DD=TMPXML
FILTER REJECT
IF (JCLSOURCE = 'PDS' AND JCLDD = 'TESTJCL') ACCEPT
CHANGE ONLY
'//JCL["./@SOURCENAME"="PDS"]/@DD' = 'PRODJCL'
$$
IMPORT EMR UPDATE DD=TMPXML
$$
```

This example illustrates exporting job events that have JCL stored in PDS datasets and then importing the events so that the JCL now is imported as Zeke JCL (e.g., when moving between a test and a production environment).

```
...
EXPORT EMR DD=TMPXML JCL=PDS
FILTER ACCEPT
IF (JOB != 1) THEN REJECT
IF ('//JCL["./@SOURCENAME"="PDS"]/@ISACTIVE'='0') THEN REJECT
CHANGE ALL
'//JCL["./@SOURCENAME"="ZEKE JCL"]/@ISACTIVE' = '1'
'//JCL["./@SOURCENAME"="PDS"]/@ISACTIVE' = '0'
$$
IMPORT EMR ADD DD=TMPXML
$$
```

Multiple Change Rule Processing

If you need to specify multiple change rules in a single CHANGE operation, consider these guidelines for how rules are processed.

An individual field value can be changed only once per operation. The first rule that specifies a change to a field value is processed. Any subsequent rules that would indicate a change to the same field value are ignored.

For example, suppose an XML record has these attributes:

```
ENAME="PROJECT"
SYS="TEST"
GROUP="ABC"
```

Example 1:

To process the record with these change rules:

```
CHANGE ONLY
IF (SYS="TEST") THEN SYS="PROD",GROUP="PRD"
IF (ENAME="PROJECT") THEN SYS="PDEV"
```

The resulting attribute values are:

```
ENAME="PROJECT"
SYS="PROD"
GROUP="PRD"
```

Although this record matches both of the change rules, only the first rule is processed and the subsequent change rule is ignored so that the same value is not changed more than once for this operation.

Example 2:

If the same XML record is processed with the same change rules, but in reverse order:

```
CHANGE ONLY
IF (ENAME="PROJECT") THEN SYS="PDEV"
IF (SYS="TEST") THEN SYS="PROD",GROUP="PRD"
```

The resulting attribute values are:

```
ENAME="PROJECT"
SYS="PDEV"
GROUP="ABC"
```

This time, the second change rule was not processed because the SYS value had already been changed to PDEV by the first change rule, so it no longer matched the SYS="TEST" criterion of the second rule.

Example 3:

In this example, let's change the second rule for processing the same record:

```
CHANGE ONLY
IF (ENAME="PROJECT") THEN SYS="PDEV"
IF (SYS="PDEV") THEN SYS="PROD",GROUP="PRD"
```

The resulting attribute values are:

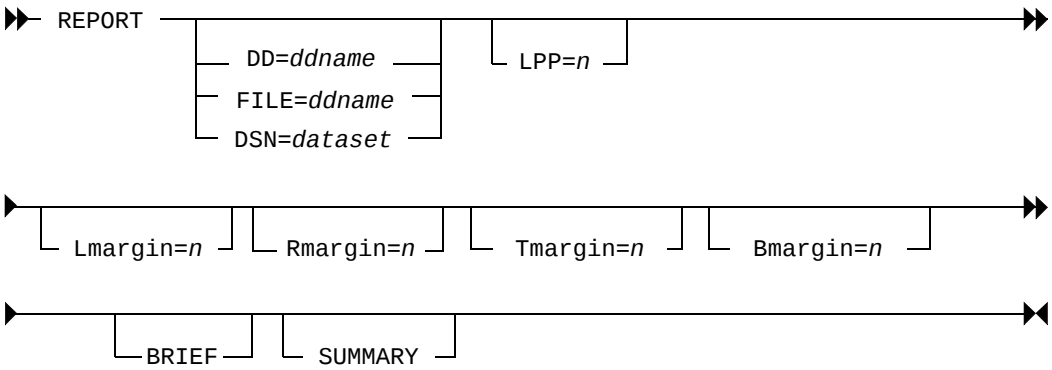
```
ENAME="PROJECT"
SYS="PDEV"
GROUP="PRD"
```

This time, the first rule changes the SYS value to PDEV. Although the second rule matches the PDEV value, the second rule cannot change the SYS value again. Only the GROUP value is changed.

REPORT—Creating Import/Export Reports

The REPORT control statement directs the report for the import/export operation to a DD name or dataset. If you omit the REPORT statement from the control statements for the import/export, then the report will be written to the SYSPRINT DD name specified in the JCL.

Syntax



Parameters

These are the valid parameters for the REPORT function:

Parameter	Description
`DD=`*`ddname`* `FILE=`*`ddname`*	Either of these parameters specifies the DD name of the dataset to which to send the report output.
`DSN=`*`dataset`*	Specifies the dataset to which to send the report output.
`LPP=`*`n`*	Specifies the number of lines per page. The default value is `66`.
`Lmargin=`*`n`*	Specifies the left margin (i.e., the number of spaces to precede each report line). The default value is `0`.
`Rmargin=`*`n`*	Specifies the right margin (i.e., number of spaces at the end of each line). The default value is `0`.

Note:

Report line length depends on the LRECL of the dataset that will receive the report output. If the report dataset has a logical record length of zero, or if the dataset has an undefined record format and the block size is the same as the logical record length, then the report will have a line length of 120.

Parameter	Description
`Tmargin=`*n*	Specifies the top margin (i.e., number of blank lines at the top of the report). The default value is `0`.
`Bmargin=`*n*	Specifies the bottom margin (i.e., number of blank lines at the bottom of the report). The default value is `0`.
`BRIEF`	Suppresses multi-line detail from a report; instead, a single detail line is printed for each event or variable that is imported or exported. **Note:** Since the EXPORT function does not create multi-line report detail, the BRIEF option has no effect on EXPORT reporting.
`SUMMARY`	Suppresses all report detail lines so that the report contains only the headings and event/variable totals. Because this option improves the overall run time, ASG recommends you use this option when detail information is not needed. **Note:** Because the SUMMARY option does not write any detail lines, you will not be able to use the detail lines as an indication of the progress of the import/export.

Report Examples

This section illustrates some sample import/export reports.

Sample EMR Report

This page includes the export specifications:

```
Zeke Z620A000            Import/Export Utility                Page: 1
Oasis X320A000                                       January 21, 2020

Export Environment
  record types        EMR
  subsys              ssss
  zeke                Z620A000
  oasis               X320A000
  version             6.2
  plex                pppppppp
  database            bbbbbbbb
  jcl sources         ZEKE


Options:

  FILTER REJECT

  IF ("descendant::emrBaseData/@applID" == "XXXX") THEN ACCEPT

  CHANGE ONLY

  IF ("descendant::emrBaseData/@applID" == "*") THEN
      "descendant::emrBaseData/@applID" = "YYYY"

Session settings:
  session id          0BA8453E01E075C87B6E1AF36D28E3BB
  session sec handle  27046B90
  security userid     "xxxxxxxx"  this is a client session

Subsys: ssss  PlexID: pppppppp  UserID: xxxxxxxx
```

This page includes the results of the export operation:

```
Zeke Z620A000            Import/Export Utility                Page: 2
Oasis X320A000                                       January 21, 2020

Import file: xxxxxxxxxxxxxxxxxxxxxxxxx      Action=ADD

Event ID  Event Name  Result
--------------------  ----------------------------
..    .
..    .
..    .
..    .

Subsys: ssss  PlexID: pppppppp  UserID: xxxxxxxx
```

Sample VAR Report

This page includes the export specifications:

```
Zeke Z620A000              Import/Export Utility                  Page: 1
Oasis X320A000                                          January 21, 2020

Export Environment
  record types        VAR
  subsys              ssss
  zeke                Z620A000
  oasis               X320A000
  version             6.2
  plex                pppppppp
  database            bbbbbbbb
  jcl sources


Options:

FILTER REJECT

IF ("descendant::varData/@app" == "XXXX") THEN ACCEPT

CHANGE ONLY

IF ("descendant::varData/@varName" == "*") THEN
    "descendant::varData/@app" = "YYYY"

Session settings:
  session id          0BA8453E01E075C87B6E1AF36D28E3BB
  session sec handle  27046B90
  security userid     "DV001"  this is a client session

Subsys: ssss  PlexID: pppppppp  UserID: DV001
```

This page includes the results of the export operation:

```
Zeke Z620A000              Import/Export Utility                  Page: 2
Oasis X320A000                                          January 21, 2020

Export file: SYS05294.T090448.RA000.MDRCXUTL.WORK.H05

Var Name          Result
---------------   ---------------------------------------
$TESTJ            Exported with changes
$TESTK            Exported with changes
$TEST             Exported with changes
$REXX1            Rejected by filter rule
$REXX2            Exported with changes
$REXX3            Exported with changes
$REXX4            Exported with changes
$X                Exported with changes
$ED               Exported with changes
$CAG              Exported with changes
$TEST2            Exported with changes
$TESTA            Exported with changes

VAR objects written        12
VAR objects rejected        1

Total objects written      12
Total objects rejected      1

Subsys: ssss  PlexID: pppppppp  UserID: DV001
```

This page includes the import specifications:

```
Zeke Z620A000              Import/Export Utility                  Page: 3
Oasis X320A000                                          January 21, 2020

Import Environment
  subsys              ssss
  zeke                Z620A000
  oasis               X320A000
  version             6.2
  plex                PPPPPPPP
  database            bbbbbbbb


Options:

Session settings:
  session id          0BA8453E01E075C87B6E1AF36D28E3BB
  session sec handle  27046B90
  security userid     "DV001"  this is a client session

Subsys: ssss  PlexID: pppppppp  UserID: DV001
```

The remaining pages include the results of the import and update operations:

```
Zeke Z620A000              Import/Export Utility                  Page: 4
Oasis X320A000                                          January 21, 2020

Import file: SYS05294.T090448.RA000.MDRCXUTL.WORK.H05        Action=UPDATE

Variable Name     Result
----------------  ---------------------------------------
$TESTJ            Variable $TESTJ Updated
$TESTK            Variable $TESTK Updated
$TEST             Variable $TEST Updated
$REXX2            Variable $REXX2 Updated
$REXX3            Variable $REXX3 Updated
$REXX4            Variable $REXX4 Updated
$X                Variable $X Updated
$ED               Variable $ED Updated
$CAG              Variable $CAG Updated
$TEST2            Variable $TEST2 Updated
$TESTA            Variable $TESTA Updated

Subsys: ssss  PlexID: PPPPPPPP  UserID: DV001
```

```
Zeke Z620A000              Import/Export Utility                  Page: 5
Oasis X320A000                                          January 21, 2020


VAR objects processed       11

VAR objects rejected         0

Total objects processed     11

Total objects rejected       0

Subsys: ssss  PlexID: pppppppp  UserID: DV001
```

Sample CAL Report

This page includes the export specifications:

```
Zeke Z620A000              Import/Export Utility                  Page: 1
Oasis X320A000                                          January 21, 2020

Export Environment
  record types        CAL
  subsys              ssss
  zeke                Z620A000
  oasis               X320A000
  version             6.2
  plex                pppppppp
  database            bbbbbbbb
  jcl sources


Options:

FILTER REJECT

IF ("descendant::calendar/@year" == "2016") THEN ACCEPT

CHANGE ONLY

IF ("descendant::calBaseData/@expireDate" == "12/31/2016") THEN
    "descendant::calBaseData/@expireDate" == "01/06/2020"

Session settings:
  session id          0BA8453E01E075C87B6E1AF36D28E3BB
  session sec handle  27046B90
  security userid     "DV001"  this is a client session

Subsys: ssss  PlexID: pppppppp  UserID: DV001
```

This page includes the results of the export operation:

```
Zeke Z620A000              Import/Export Utility                  Page: 2
Oasis X320A000                                          January 21, 2020

Export file: SYS05294.T090448.RA000.MDRCXUTL.WORK.H05

Cal Name          Result
---------------  ---------------------------------------
SPECIAL           Exported with changes
USERACCT          Exported with changes
STANDARD          Exported with changes
ASGTEST           Rejected by filter rule
ASG               Exported with changes

CAL objects written         5
CAL objects rejected        1

Total objects written       5
Total objects rejected      1

Subsys: ssss  PlexID: PPPPPPPP  UserID: DV001
```

This page includes the import specifications:

```
Zeke Z620A000              Import/Export Utility                  Page: 3
Oasis X320A000                                           January 21, 2020

Import Environment
  subsys              ssss
  zeke                Z620A000
  oasis               X320A000
  version             6.2
  plex                PPPPPPPP
  database            bbbbbbbb


Options:

Session settings:
  session id          0BA8453E01E075C87B6E1AF36D28E3BB
  session sec handle  27046B90
  security userid     "DV001"  this is a client session

Subsys: ssss  PlexID: PPPPPPPP  UserID: DV001
```

The remaining pages include the results of the import and update operations:

```
Zeke Z620A000              Import/Export Utility                  Page: 4
Oasis X320A000                                           January 21, 2020

Import file: SYS05294.T090448.RA000.MDRCXUTL.WORK.H05        Action=UPDATE

Calendar Name     Result
---------------   ---------------------------------------
SPECIAL           Calendar SPECIAL Updated
USERACCT          Calendar USERACCT Updated
STANDARD          Calendar STANDARD Updated
ASG               Calendar ASG Updated

Subsys: ssss  PlexID: pppppppp  UserID: DV001
```

```
Zeke Z620A000              Import/Export Utility                  Page: 5
Oasis X320A000                                           January 21, 2020


CAL objects processed        4

CAL objects rejected         0

Total objects processed      4

Total objects rejected       0

Subsys: ssss  PlexID: pppppppp  UserID: DV001
```

Sample Import and Export Jobs

This section provides some examples of using the ZEKEXUTL utility program to update Zeke database records or migrate records from one database to another.

Note:

Each function requires its own, unique SYSMDUMP dataset. See the troubleshooting appendix in the *ASG-Zeke Scheduling for z/OS Installation Guide* for details on enabling IPCS readable dumps.

Note:

In these examples, temporary datasets are used (see "Temporary and Permanent Datasets" on page 118).

Sample EMR Export Jobs

Figure 3 illustrates exporting EMRs for job events to a new dataset that is cataloged and retained for processing later.

Figure 3 • Sample Job—Exporting EMRs for Job Events

```
...
//ZEKEXUTL EXEC PGM=ZEKEXUTL,REGION=0M,
// PARM='SUBSYS=SSSI,DATASPACE'
//STEPLIB  DD DISP=SHR,DSN=ZEKE.LINKLIB
//         DD DISP=SHR,DSN=OASIS.LINKLIB
//         DD DISP=SHR,DSN=RIS.LINKLIB
//*
//SYSPRINT DD SYSOUT=*
//SYSOUT   DD SYSOUT=*
//RPT1     DD SYSOUT=*,DCB=(LRECL=80,RECFM=F)
//JOBEMR   DD DISP=(NEW,CATLG),DSN=hlq.export.data,
//            DCB=(LRECL=27994,RECFM=VB,BLKSIZE=27998),
//            UNIT=SYSDA,SPACE=(CYL,(50,50),RLSE)
//SYSMDUMP DD DISP=SHR,DSN=your.dump.dataset
//SYSIN    DD *
* This sample exports EMR records that are for JOB events
* to a dataset that will be cataloged and retained.
REPORT DD=RPT1 LPP=55 L=3 R=3 T=1 B=1
EXPORT EMR DD=JOBEMR
FILTER REJECT
IF (JOB) THEN ACCEPT
/*
//*
//ENVIRON  DD *
TZ=EST5EDT
/*
//
```

In this example, a report will be written to DD RPT1 with 55 lines per page, a left margin of three chars, a right margin of three chars, a top margin of one line, and a bottom margin of one line. The utility will attempt to format the records to fit within the available LRECL.

Figure 4 illustrates how to export EMRs for other event types to a new dataset that is cataloged and retained for processing later. In this example, the XPath location name is used for filtering. (See Appendix A, "XPath Location Paths," on page 609 for more information.)

Figure 4 • Sample Job—Exporting EMRs for Non-job Events

```
...
//ZEKEXUTL EXEC PGM=ZEKEXUTL,REGION=0M,
// PARM='SUBSYS=SSSI,DATASPACE'
//STEPLIB  DD DISP=SHR,DSN=ZEKE.LINKLIB
//         DD DISP=SHR,DSN=OASIS.LINKLIB
//         DD DISP=SHR,DSN=RIS.LINKLIB
//*
//SYSPRINT DD SYSOUT=*
//SYSOUT   DD SYSOUT=*
//RPT1     DD SYSOUT=*,DCB=(LRECL=80,RECFM=F)
//NJOBEMR  DD DISP=(NEW,CATLG),DSN=hlq.export.data,
//            DCB=(LRECL=27994,RECFM=VB,BLKSIZE=27998),
//            UNIT=SYSDA,SPACE=(CYL,(50,50),RLSE)
//SYSMDUMP DD DISP=SHR,DSN=your.dump.dataset
//SYSIN    DD *
* This sample exports EMR records that are for Non-JOB events
* to a dataset that will be cataloged and retained.
REPORT DD=RPT1 LPP=55 L=3 R=3 T=1 B=1
EXPORT EMR DD=NJOBEMR
FILTER REJECT
IF ("descendant::emrBaseData/@event.type" > "1") THEN ACCEPT
/*
//*
//ENVIRON  DD *
TZ=EST5EDT
/*
//
```

Sample Calendar Export Job

Figure 5 illustrates exporting calendars to a new dataset that is cataloged and retained for processing later.

Figure 5 • Sample Job—Exporting Calendars

```
//DVNEKGT JOB (10074,250),'CREATE 50',
//             MSGLEVEL=(1,1),
//             MSGCLASS=X,NOTIFY=TSOUSR,
//             CLASS=A
//*
//ZEKEXUTL EXEC PGM=ZEKEXUTL,REGION=0M,
// PARM='SUBSYS=ZK62'
//STEPLIB  DD   DISP=SHR,DSN=QAL.ARL.ZEKE.R620A.SZEKLMD0
//         DD   DISP=SHR,DSN=QAL.ARL.OASIS.R320A.SALTLMD0
//         DD   DISP=SHR,DSN=RIS.PROD.R212A.LINKLIB
//SYSPRINT DD   SYSOUT=*,DCB=(LRECL=133,RECFM=F)
//SYSOUT   DD   SYSOUT=*,DCB=(LRECL=133,RECFM=F)
//RPT1     DD   SYSOUT=*,DCB=(LRECL=121,RECFM=FA)
//CALOUT   DD   DISP=(NEW,KEEP),DSN=TSOUSR.CAL62.XML,
//         UNIT=SYSDA,SPACE=(CYL,(1)),
//         DCB=(LRECL=6144,RECFM=VB,BLKSIZE=27998)
//SYSMDUMP DD DISP=SHR,DSN=your.dump.dataset
//SYSIN    DD   *
EXPORT CAL DD=CALOUT
$$
//ENVIRON  DD *
TZ=EST5EDT
//
```

Sample Calendar Import Job

Figure 6 illustrates exporting calendars to a new dataset that is cataloged and retained for processing later.

Figure 6 • Sample Job—Exporting Calendars

```
//DVNEKGT JOB (10074,250),'CREATE 50',
//             MSGLEVEL=(1,1),
//             MSGCLASS=X,NOTIFY=TSOUSR,
//             CLASS=A
//*
//ZEKEXUTL EXEC PGM=ZEKEXUTL,REGION=0M,
//    PARM='SUBSYS=ZK62'
//STEPLIB  DD   DISP=SHR,DSN=QAL.ARL.ZEKE.R620A.SZEKLMD0
//         DD   DISP=SHR,DSN=QAL.ARL.OASIS.R320A.SALTLMD0
//         DD   DISP=SHR,DSN=RIS.PROD.R212A.LINKLIB
//SYSPRINT DD   SYSOUT=*,DCB=(LRECL=133,RECFM=F)
//SYSOUT   DD   SYSOUT=*,DCB=(LRECL=133,RECFM=F)
//RPT1     DD   SYSOUT=*,DCB=(LRECL=121,RECFM=FA)
//CALIN    DD   DISP=SHR,DSN=TSOUSR.CAL62.XML
//SYSMDUMP DD DISP=SHR,DSN=your.dump.dataset
//SYSIN    DD   *
IMPORT CAL ADD DD=CALIN
$$
//ENVIRON  DD *
TZ=EST5EDT
```

Sample Mass Record Change Jobs

Each sample job illustrated in this section performs all of these operations:

- Exports records to the work (temporary) file.
- Specifies that each exported record should contain only the information to be changed and any required fields.
- Imports the exported records from the work file using the UPDATE option.

Figure 7 performs the mass change using a single job step:

Figure 7 • Sample Job—Applying Mass Changes to VAR Records in One Step

```
//*
//ZEKEXUTL EXEC PGM=ZEKEXUTL,REGION=0M,
// PARM='SUBSYS=SSSI,DATASPACE'
//STEPLIB  DD DISP=SHR,DSN=ZEKE.LINKLIB
//         DD DISP=SHR,DSN=OASIS.LINKLIB
//         DD DISP=SHR,DSN=RIS.LINKLIB
//*
//SYSPRINT DD SYSOUT=*
//SYSOUT   DD SYSOUT=*
//RPT1     DD SYSOUT=*,DCB=(LRECL=80,RECFM=F)
//WORK     DD DISP=NEW,DSN=&&WORK,UNIT=SYSDA,
//            DCB=(LRECL=27994,RECFM=VB,BLKSIZE=27998),
//            SPACE=(TRK,(10,10),RLSE)
//SYSMDUMP DD DISP=SHR,DSN=your.dump.dataset
//SYSIN    DD *
* This sample changes all variables for the "XXXX" application
* to "YYYY" using a work file for the filtered records.
REPORT DD=RPT1 LPP=55 L=3 R=3 T=1 B=1
EXPORT VAR DD=WORK
FILTER REJECT
IF (APPL='XXXX') THEN ACCEPT
CHANGE ONLY
IF (NAME=*) THEN APPL="YYYY"
$$
* We now import the records we just exported which will update
* the application name with "YYYY"
IMPORT VAR UPDATE DD=WORK
$$
/*
//*
//ENVIRON  DD *
TZ=EST5EDT
/*
//
```

Figure 8 performs the mass change using two job steps:

Figure 8 • Sample Job—Applying Mass Changes to VAR Records in Two Steps

```
...
//EXPORT   EXEC PGM=ZEKEXUTL,REGION=0M,
// PARM='SUBSYS=SSSI,DATASPACE'
//STEPLIB  DD DISP=SHR,DSN=ZEKE.LINKLIB
//         DD DISP=SHR,DSN=OASIS.LINKLIB
//         DD DISP=SHR,DSN=RIS.LINKLIB
//*
//SYSPRINT DD SYSOUT=*
//SYSOUT   DD SYSOUT=*
//RPTEXP   DD SYSOUT=*,DCB=(LRECL=80,RECFM=F)
//WORK     DD DISP=(NEW,PASS),DSN=&&WORK,UNIT=SYSDA,
//            DCB=(LRECL=27994,RECFM=VB,BLKSIZE=27998),
//            SPACE=(TRK,(10,10),RLSE)
//SYSMDUMP DD DISP=SHR,DSN=your.dump.dataset
//SYSIN    DD *
* This sample changes all variables for the "XXXX" application
* to "YYYY" using a work file for the filtered records.
REPORT DD=RPTEXP LPP=55 L=3 R=3 T=1 B=1
EXPORT VAR DD=WORK
FILTER REJECT
IF (APPL='XXXX') THEN ACCEPT
CHANGE ONLY
IF (NAME=*) THEN APPL="YYYY"
/*
//*
//ENVIRON  DD *
TZ=EST5EDT
/*
//IMPORT   EXEC PGM=ZEKEXUTL,REGION=0M,COND=(0,NE,EXPORT),
// PARM='SUBSYS=SSSI,DATASPACE'
//STEPLIB  DD DISP=SHR,DSN=ZEKE.LINKLIB
//         DD DISP=SHR,DSN=OASIS.LINKLIB
//         DD DISP=SHR,DSN=RIS.LINKLIB
//*
//SYSPRINT DD SYSOUT=*
//SYSOUT   DD SYSOUT=*
//RPTIMP   DD SYSOUT=*,DCB=(LRECL=80,RECFM=F)
//WORK     DD DISP=(OLD,DELETE),DSN=*.EXPORT.WORK
//SYSMDUMP DD DISP=SHR,DSN=your.dump.dataset
//SYSIN    DD *
* We now import the records we just exported which will update
* the application name with "YYYY"
REPORT DD=RPTIMP LPP=55 L=3 R=3 T=1 B=1
IMPORT VAR UPDATE DD=WORK
/*
//*
//ENVIRON  DD *
TZ=EST5EDT
/*
//
```

Moving from a Test to a Production System

This sample job illustrates how to move records from a test environment to a production environment using the import/export utility.

In Figure 9, the Zeke in the test environment has the subsystem name TEST. The production Zeke has the subsystem name PROD.

Figure 9 • Sample Job—Moving Records from Test to Production

```
...
//EXPORT  EXEC PGM=ZEKEXUTL,REGION=0M,PARM='SUBSYS=TEST'
//STEPLIB  DD DISP=SHR,DSN=ZEKE.LINKLIB
//         DD DISP=SHR,DSN=OASIS.LINKLIB
//         DD DISP=SHR,DSN=RIS.LINKLIB
//*
//SYSPRINT DD SYSOUT=*
//SYSOUT   DD SYSOUT=*
//RPTEXP   DD SYSOUT=*
//WORKEMR  DD DISP=(NEW,PASS),DSN=&&WORKEMR,UNIT=SYSDA,
//            DCB=(LRECL=27994,RECFM=VB,BLKSIZE=27998),
//            SPACE=(CYL,(50,50),RLSE)
//WORKVAR  DD DISP=(NEW,PASS),DSN=&&WORKVAR,UNIT=SYSDA,
//            DCB=(LRECL=27994,RECFM=VB,BLKSIZE=27998),
//            SPACE=(TRK,(10,10),RLSE)
//SYSMDUMP DD DISP=SHR,DSN=your.dump.dataset
//SYSIN    DD *
* This sample exports EMR records for the "TESTGL" application
* in preparation for moving to the production Zeke scheduler.
* We use the DATASPACE option on the EXPORT since we didn't
* include it in the parms for ZEKEXUTL.
REPORT DD=RPTEXP LPP=55 L=3 R=3 T=1 B=1
EXPORT EMR DD=WORKEMR DATASPACE
FILTER REJECT
IF (APPL='TESTGL') THEN ACCEPT
$$
* And we want any VAR records that are for the "TESTGL"
* application as well.
EXPORT VAR DD=WORKVAR
FILTER REJECT
IF (APPL='TESTGL') THEN ACCEPT
/*
//*
//ENVIRON  DD *
TZ=EST5EDT
/*
//IMPORT   EXEC PGM=ZEKEXUTL,REGION=0M,COND=(0,NE,EXPORT),
// PARM='SUBSYS=PROD,DATASPACE'
//STEPLIB  DD DISP=SHR,DSN=ZEKE.LINKLIB
//         DD DISP=SHR,DSN=OASIS.LINKLIB
//         DD DISP=SHR,DSN=RIS.LINKLIB
//*
//SYSPRINT DD SYSOUT=*
//SYSOUT   DD SYSOUT=*
//RPTIMP   DD SYSOUT=*
//WORKEMR  DD DISP=(OLD,DELETE),DSN=*.EXPORT.WORKEMR
//WORKVAR  DD DISP=(OLD,DELETE),DSN=*.EXPORT.WORKVAR
```

```
//SYSMDUMP DD DISP=SHR,DSN=your.dump.dataset
//SYSIN    DD *
* We now import the records we just exported.
* We also change the application name from "TESTGL"
* to "GL123" as part of the importing of these records.
REPORT DD=RPTIMP LPP=55 L=3 R=3 T=1 B=1
IMPORT EMR ADD DD=WORKEMR
CHANGE ALL
IF (APPL="TESTGL") THEN APPL="GL123"
$$
IMPORT VAR ADD DD=WORKVAR
CHANGE ALL
IF (APPL="TESTGL") THEN APPL="GL123"
/*
//*
//ENVIRON  DD *
TZ=EST5EDT
/*
//
```

3 ZEKESET Utility

You can control jobstream flow by using the ZEKESET utility to perform these tasks:

- Set variables
- Set the step condition code
- Set a user abend code
- Execute Zeke operator commands
- Execute z/OS commands, JES commands, or VM commands.

This chapter contains these topics:

Topic	Page
Using ZEKESET	170
SET Statement	170
SET ABEND—Terminate a Job	175
SET ROUTE—Create Large Block Letter Separator Pages	175
SET xCOM—Issue System and Zeke Commands	176
SET VARIABLE—Define Zeke Variables	179
SET COMMENT—Add Comment Lines	181
SET CONDCODE/RETCODE—Set Condition/Return Codes	181
SET GOTO—Branch within Control Statements	183
SET WAIT—Set a Wait Interval	184
OPTION—Set ZEKESET Execution Options	185
CDATE—Perform Date Calculations	186
ON ERROR—Act on Syntax Errors	195

Using ZEKESET

To use the program features, you execute the ZEKESET utility in any job stream. ZEKESET functions can be made conditional by using the IF clause of the SET statement.

Control statements follow the same syntax rules as Zeke batch utility input statements (refer to "Batch Utility Commands" on page 2).

To use the ZEKESET program

- Execute this program in any jobstream to execute the job control SET statements:

```
//ZSET     JOB   ,ZEKESET.SAMPLE,CLASS=A
//ZSAMPLE  EXEC  PGM=ZEKESET,PARM='SUBSYS=SSSI'
//STEPLIB  DD   DSN=ZEKE.LINKLIB,DISP=SHR
//         DD   DSN=OASIS.LINKLIB,DISP=SHR
//SYSPRINT DD    SYSOUT=A
//SYSIN    DD    *
  SET ROUTE (SAMPLE ROUTING PAGE)
  SET ABEND 4 IF $ABENDCODE GT 1
/*
```

The control statements are read from SYSIN and are listed along with informational and error messages using the ddname SYSPRINT.

Note:

You can use the OPTION NOSUB statement (see "OPTION—Set ZEKESET Execution Options" on page 185) to prevent variable substitution in ZEKESET statements.

SET Statement

This section explains basic SET statement usage.

Statement Syntax

SET statements follow the same syntax rules as the ZEKE batch utility input statements (see "Batch Utility Commands" on page 2 for details).

- Separate parameters and operands by either an equal sign (=) or by one or more spaces.

- Separate multiple operands within parentheses by a comma or by one or more spaces.
- Separate parameters by a comma or by one or more spaces.

Note:

Use either a comma or one or more spaces to separate items, but not both. A comma followed by a space indicates end of statement to Zeke.

- Precede comments with a comma and at least one space. For example:

```
SET VAR $LS EQ STEP1 ,
```
Note comma prior to this comment.

```
SET VAR $ABC EQ OK ,
```
Any Zeke statement can have a comment.

```
SET ROUTE (WORD1 WORD2 ,
```
A comma must precede the comment.

- Complete each ZEKESET statement on a single line (columns 1 through 72). Continuation statements are not supported by the ZEKESET program.

Up to three consecutive lines (up to 216 characters in total) can be concatenated for a SET ZCOM statement. The first line must specify a SET ZCOM statement. Subsequent lines are concatenated if the first word of each line *is not* one the these reserved words:

```
CDATE
CD
ON
OPTION
SET
*
label
```

where:

label is a string up to eight characters long followed by an asterisk (*).

any line beginning with any number of asterisks is considered a comment.

For example:

```
SET ZCOM 'ZALTER EVENT 32 CONTROL=NX PRIORITY=1 DPRI=51 TIMES=1
FREQCALC=CLOCK EARLY=0900 LATEEND=1100 MUSTEND=1200 NOTAFTER=1300
SCHED=0930 NEWAPPL=APL45678 NEWGRP=AFG TRIGOPT=FIRST'
```

Spaces are preserved. To split a word across lines, the word must extend to column 72 and continue in column 1 of the next line.

Note:

Only the first line of a SET ZCOM state is written to the job log; however, the entire command is written to the OASIS data space log.

- If you are running multiple versions of Zeke, the EXEC statement must include the subsystem name, if other than SSSI.

  ```
  //Z1 EXEC  PGM=ZEKESET,PARM='SUBSYS=subsystem name'
  ```

- Comments and blank lines are supported. Any line with an asterisk as the first non-space character is considered a comment line. Comment lines and blank lines are printed, and are otherwise ignored. Comments can also be placed after a comma on the same line as a ZEKESET input statement. For example:

  ```
  CDATE OUTDATE EQ DATE,          Set to current date
  ```

- If desired, you can indent input lines and comment lines for better legibility.

IF Clause of the SET Statement

The IF clause of the Zeke SET statement makes the statement conditional. When the IF clause is true, Zeke processes the SET statement. When the IF clause is false, the SET statement is bypassed. This is the format of the IF clause:

```
IF DATA1 OPERATOR DATA2
```

where *OPERATOR* can be any of these values:

Operator	Meaning
EQ	EQual
LE	Less than or Equal
LT	Less Than
GT	Greater Than
GE	Greater than or Equal

Operator	Meaning
NE	Not Equal

DATA1 and DATA2 refer to the "Special Names" on page 173. These special names are referenced by several commands. IF clauses on SET statements can check certain special names in addition to checking variables.

Examples

These are sample conditional SET statements:

```
SET VAR $XYZ EQ NO IF $ABC EQ YES
SET VCOM 'CP DRAIN 00E' IF ZEKECPU EQ A
SET CANCEL IF PARTN NE BG
SET CONDCODE 4 IF ZEKECPU EQ A
SET ABEND 010 IF LASTCOND GT 4
SET VAR $STPNAME EQ WKLY003 IF $VARNAME1 LE 57
SET CONDCODE 20 IF TIME LE 080000
SET VAR $PR01STP EQ OK IF $PRFLAG EQ NORMAL
SET VAR $BADNEWS EQ *JOB-CANCEL* IF ABTYPE NE NO
SET GOTO BYPSTMTS IF ABTYPE EQ SYSTEM
```

Special Names

DATA1 and DATA2 of an IF clause can be any of the special names listed below; however, the value for DATA1 must have the same data type as the value for DATA2. For example, if the special name is numeric only for DATA1, then the special name must be numeric only for DATA2. Special names cannot be used as the object of SET statements.

Note:

The difference between special names and Zeke variables is that special names are predefined to Zeke, while Zeke variables are user-defined and begin with a dollar sign ($).

Special Name	Data Type	Description
ABCODE	N	Abend code value, if an abend occurred.
ABTYPE	A/N	Abend type, if an abend occurred: NO USER SYSTEM FLUSH

Special Name	Data Type	Description
COMRC	N	Return code from CP after issuing a SET VCOM statement.
CPUID	A/N	8-digit CPU serial number (e.g., `12345678`).
DATE	A/N	Current system date, in either *`mm/dd/yyyy`* or *`dd/mm/yyyy`* format. **Note:** DATE cannot be used in greater than or less than comparisons because the year is not compared first. For example, if DATE contains 12/15/2016 and is compared to 02/02/2020, the result is that the value contained in DATE is greater than 02/02/2020 because 12/15 is greater than 02/02. Use the DATEJ special name to compare dates.
DATEJ	N	Julian system date, in *`yyyyddd`* format.
DAY	N	Day of the week. The valid values range from `1` (Monday) through `7` (Sunday).
EVENT	N	Event number of this job, if it is a Zeke job event.
HIGHCOND	N	Numeric value of the highest condition code of any step in this job.
JOBNAME	A/N	Name of the current job.
LASTCOND	N	Value of the condition code for the previous job step.
LASTPGM	A/N	Name of the program executed by the previous job step.
LASTSTEP	A/N	Name of the previous job step.
THISPGM	A/N	Name of the program executed by this job step.
THISSTEP	A/N	Name of the current job step.
TIME	N	Current system time (in *`HHMMSS`* format). If the hour is less than `10`, the leading zero is dropped.
VERSION	N	Number (up to 5 digits long) that identifies a particular version of the event.
ZEKECPU	A/N	Zeke CPU name for this system.
$XXXXX	A/N	*XXXXX* represents a Zeke variable.

Special Name	Data Type	Description
LITERAL	A/N	Any character string up to 64 bytes that is delimited.
NNN	N	Any explicitly stated numeric value.
XXX	A/N	Any explicitly stated character value that contains no special characters.

SET ABEND—Terminate a Job

The SET ABEND statement immediately terminates a job with any user abend code. No more control statements in the input stream are executed. For example:

```
SET ABEND 1,
     User 1 Abend.
```

SET ROUTE—Create Large Block Letter Separator Pages

The SET ROUTE statement generates large block letter separator pages. Separator pages are produced on the dataset allocated to SYSPRINT.

ROUTE specifies the word or words to be printed in large block letters. You can enter up to ten words, and each word can be up to eight characters long. Each word specified is printed in large block letters on the generated separator page separated by three blank lines. (Each word is printed on the same page).

Enter multiple routing words within a single set of parentheses. For example:

```
SET ROUTE (WORD1 WORD2 WORD3 ...  )
```

Add this parameter to print the same separator page multiple times.

Parameter	Description
COPY	Print the same separator page up to ten times in a row. For example, this statement prints two copies of a routing page with the jobname in large block letters: `SET ROUTE JOBNAME COPY 2`

Zeke variables and special names can be used as the operand of the ROUTE parameter. When Zeke variables are used, the value of the variable is the word that is printed in large block letters. When Zeke special names (e.g., ZEKECPU, COMRC, DATEJ, ABTYPE, ABCODE, etc.) are used, the value of the special name is printed.

Character values are truncated to the first eight characters, and numeric values are always considered to be 8-digit decimal numbers.

Large block letters are printed for letters (i.e., A through Z), digits 0 through 9, and these special characters: dollar sign (`$`), asterisk (`*`), dot (`.`), hyphen/dash (`-`), and slash (`/`). Other special characters are treated as blanks.

Examples

This statement specifies one routing word:

```
SET ROUTE DATACTL
```

This statement specifies two routing words:

```
SET ROUTE (DATA CONTROL)
```

This statement specifies four routing words:

```
SET ROUTE (Give to sys progrmr)
```

This statement specifies special names for two routing words:

```
SET ROUTE (Jobname this step)
```

This statement specifies a variable for a routing word:

```
SET ROUTE $VAR1
```

This statement specifies a combination of a special name, variable, and a literal:

```
SET ROUTE (JOBNAME $VAR1 **SAVE**)
```

This statement prints two copies of a routing page with the jobname in large block letters:

```
SET ROUTE JOBNAME COPY 2
```

SET xCOM—Issue System and Zeke Commands

The SET statement can issue z/OS commands, JES commands, Zeke operator commands, VM CP commands. The commands are issued by Zeke as the SET statement is processed.

These are the requirements for the SET xCOM statement:

- Each statement must specify one of these codes followed by the command or response:

Code	Meaning
SCOM	SJES or system command.
VCOM	VM command/CP command. See “Issuing CP Commands” on page 179 for additional considerations.
ZCOM	Zeke command. See “Issuing Zeke Commands” on page 178 for additional considerations.

- The statement must be enclosed in delimiters.

Examples

This statement issues the system command DISPLAY ALL:

```
SET SCOM 'D A,L'
```

This statement issues a system command (JES2 pause of initiator):

```
SET SCOM '$PI I4'
```

This statement sends a CP message to an operator:

```
SET VCOM 'MSG OPERATOR HI THERE'
```

This statement issues a Zeke operator command:

```
SET ZCOM 'ZREFRESH JOB TSTA0500'
```

This statement executes a Zeke operator command to request the system ID, initiator mapping, initiator availability:

```
SET ZCOM 'ZID ZMAP ZD AV'
```

This statement sends a VM message to a CMS user:

```
SET VCOM 'M BOB HAVE A NICE DAY'
```

The IF clause makes the SET statement conditional. For example:

```
SET SCOM ’$TI I1,A’ IF ZEKECPU EQ A
```

Issuing Zeke Commands

You can issue Zeke commands through either the ZCOM or SCOM function:

- Recommended. The SET ZCOM function executes the Zeke command in the batch job that runs the ZEKESET program. You can schedule the SET ZCOM function in a Zeke job event.
- The SET SCOM function issues the Zeke command to the operator console. The Zeke command is executed through the Zeke command processor subtask.

Note:

While the SET SCOM function produces the same result as SET ZCOM, using this method of command processing can impact performance.

See “Issuing Zeke Operator Commands” on page 255 for more information on Zeke operator commands.

See "Command Events" in Chapter 4, "Events" in the *ASG-Zeke Scheduling for z/OS User's Guide* for more information on defining command events that issue Zeke commands.

Issuing CP Commands

When Zeke issues a CP command from a SET VCOM statement and encounters a nonzero return code from CP, Zeke waits one second and issues the command again. Zeke retries the command ten times, issues a message to the operator specifying the return code from CP, and bypasses the SET VCOM statement.

To modify the number of retry attempts, use the RETRY parameter followed by the number (ranging from `1` through `10`) of retry attempts. For example:

```
SET VCOM 'SMSG USERID HI THERE' RETRY 3    retry 3 times
```

To check the CP return code after a set VCOM statement is issued, use the IF clause with the special name COMRC. The value of COMRC is the CP return code provided to Zeke.

SET VARIABLE—Define Zeke Variables

The SET VARIABLE statement defines Zeke variables and changes existing variable values. If a variable in a SET statement is not already defined in the database, it is added and processing continues. See the *ASG-Zeke Scheduling for z/OS User's Guide* for additional information on using variables.

A variable can be set to any of these values:

- Another Zeke variable
- A numeric value (up to eight places) explicitly stated (any leading zeros are truncated)
- A character value (up to 64 characters long) explicitly stated or contained in delimiters (any trailing blanks are truncated)
- One of the special names supported by the IF clause of the SET statement, such as COMRC

The variable assumes the format, numeric or character, of the value to which it is set. When a variable is set to a numeric value, it can be set to plus and/or minus another value. This is done by using the operators (+) and (-) to add and subtract other values, respectively. A space must precede and follow either symbol.

Note:

You can use both a plus (+) and a minus (-) on the same statement. However, only one plus and one minus can be used per statement. For example:

```
SET VAR $VAR1 EQ $A + $B - $C
```
This statement is valid.

```
SET VAR $VAR1 EQ $A + $B + $C
```
This statement is not valid.

Examples

This statement sets a variable in alphanumeric format:

```
SET VAR $VARNAME EQ OPER1,
```

This statement sets a variable in numeric format:

```
SET VAR $VARNAME EQ OPER1 + OPER2 - OPER3,
```

This statement sets variable to the character value `NO`:

```
SET VAR $XYZ EQ NO,
```

This statement sets variable to the character value `YES`:

```
SET VAR $OPERFLAG EQ 'YES',
```

This statement sets variable to the numeric value `400`:

```
SET VAR $CTR1 EQ 400,
```

This statement adds 1 to the value of $CTR1 and sets $CTR2 to that numeric value:

```
SET VAR $CTR2 EQ $CTR1 + 1,
```

This statement increments the variable value by 1:

```
SET VAR $CTR EQ $CTR + 1,
```

This statement adds values of $A2 and $A3, subtracts that sum from $A4, and sets $A1 to that result:

```
SET VAR $A1 EQ $A2 + $A3 - $A4,
```

This statement adds values of $B1 and $B2, subtracts 1 from that sum, and sets $B1 to that result:

```
SET VAR $B1 EQ $B1 + $B2 - 1,
```

This statement sets variable $KAM to the event number of the event that set the variable:

```
SET VAR $KAM EQ EVENT,
```

SET COMMENT—Add Comment Lines

You can document the processes you have applied in the ZEKESET execution. Use the COMMENT parameter to specify from 1 through 40 bytes of comments per SET COMMENT statement. For example:

```
SET COMMENT (THIS PROGRAM EXECUTES DAILY)
```

SET CONDCODE/RETCODE—Set Condition/Return Codes

The SET CONDCODE and SET RETCODE statements have the same function.

Use the SET CONDCODE or SET RETCODE statement to set condition or return codes within the ZEKESET program.

The ZEKESET job step condition code, sometimes called the return code or completion code, is normally set to zero by the ZEKESET program. However, if errors are detected in the input control statements, the condition or return code is set to `4`, `8`, or `12`, depending on the type of error. You can also use the SET CONDCODE *nnnn* statement or SET RETCODE statement to set the condition or return code to any value ranging from zero through 4095.

If a user-defined value is set, that value is used when the program terminates. ZEKESET never overrides a user-set condition or return code. In other words, if errors are detected in the input stream, but a SET CONDCODE or SET RETCODE statement was successfully processed, the normal error code is ignored and the user-defined value is used in its place.

If more than one SET CONDCODE or SET RETCODE statement is processed, the last value set is the one that is used for the step condition or return code. This allows you to set the condition or return code based on the value of other data or other conditions. For example, a condition or return code can be set by a statement in the ZEKESET input stream, then changed by one or more following statements. This feature allows control of the Zeke jobstream flow.

The ability to set the step condition or return code can be used with the Zeke special names LASTCOND and HIGHCOND. These special names represent the highest condition or return code from the last step executed and the highest condition or return code encountered in the jobstream, respectively. These names can be used in the IF portion of a Zeke job control SET statement to test the ZEKESET condition or return code or any other job step condition or return code.

For example:

```
//JSO10 EXEC PGM=ZEKESET,PARM='SUBSYS=SSSI'
   SET CONDCODE 99 IF DAY EQ 7
   SET ROUTE (ABEND) IF LASTCOND EQ 99
/*
```

The SET CONDCODE or SET RETCODE statement can be used with the z/OS job control EXEC statement parameter COND= to control the sequence of steps executed within a jobstream. Execute ZEKESET at any point within a job and set the step condition or return code to any value. Based on the current day, time, date, the CPU-ID, a variable value, the return code of a previous step, an abend code, etc., test the new condition or return code through the COND parameter of a subsequent EXEC statement to execute or bypass the remaining steps as desired.

For example, this enables a step return code completion of 12, if the variable is equal to 6:

```
//ZSAMP JOB ,ZEKESET.SAMPLE,CLASS=A
//JSO10 EXEC PGM=ZEKESET,PARM='SUBSYS=SSSI'
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
  SET RETCODE 12 IF $VARNAME EQ 6
/*
//
```

For example:

```
//ZSAMP JOB ,ZEKESET.SAMPLE,CLASS=A
//CKDAY EXEC  PGM=ZEKESET,PARM='SUBSYS=SSSI'
//SYSPRINT DD SYSOUT=A
//SYSIN    DD *
  SET RETCODE 99 IF DAY EQ 1
/*
//S1    EXEC  PGM=PROG1,COND=(99,EQ,CKDAY)
//SYSPRINT DD SYSOUT=A
//S2    EXEC  PGM=PROG2
//CKS2  EXEC  PGM=ZEKESET,PARM='SUBSYS=SSSI'
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
  SET RETCODE 12 IF LASTCOND GT 1
  SET RETCODE 0  IF LASTCOND EQ 53
/*
//S3    EXEC  PGM=PROG3,COND=(12,EQ,CKS2)
//SYSPRINT DD SYSOUT=A
//
```

In this example:

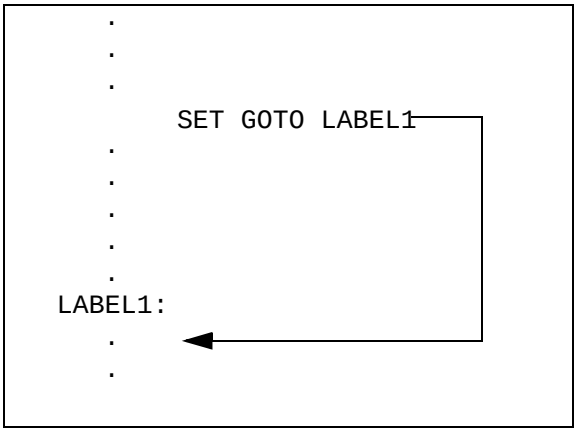
- The first job step executes the program ZEKESET. If the current day is Monday, the step terminates with a return code of 99. Otherwise, the return code is zero.
- If the current day is Monday, the second job step (S1) is not executed, because the EXEC statement COND parameter specifies to bypass the step if the return code from step CKDAY is equal to 99.
- The job step CKS2 executes ZEKESET again.
  - If the previous step (S2) has a return code greater than 1 and less than 53, the ZEKESET step is completed with a return code of 12.
  - If the previous step (S2) has a return code of 53, the ZEKESET step terminates with a return code of zero. This return code determines whether the final step, S3, is executed.

SET GOTO—Branch within Control Statements

Use the ZEKESET statement SET GOTO to bypass control statements in the input stream. This facility operates similar to branching within a program; however, SET GOTO cannot branch backwards.

The statement specifies a *label* to be searched for. Labels are coded as the first non-blank item on a control statement. The label name can be up to 64 characters long and must end with a colon (:). Statements containing labels cannot contain other SET commands. A label statement contains only the label name followed by the colon. If a specified label cannot be found, processing terminates.

All control statements between the GOTO statement and the specified label statement are bypassed as shown in this figure. This includes all conditional and unconditional SET statements, even other GOTO statements.



Any of the special names previously described can be used in a GOTO statement IF clause. For example:

```
//ZSAMP JOB ,ZEKESET.SAMPLE,CLASS=A
//S1    EXEC    PGM=PROG1
//S2    EXEC    PGM=ZEKESET,PARM='SUBSYS=SSSI'
//SYSPRINT DD   SYSOUT=A
//SYSIN DD *
    SET CONDCODE 4
    SET CONDCODE 12 IF LASTCOND GT 4
    SET GOTO STEP-WAS-OK  IF LASTCOND EQ 0
    SET ROUTE (STEP S1 HAD NON-ZERO COND CODE)
    SET GOTO CONTINUE
  STEP-WAS-OK:
    SET ROUTE (STEP S1 WAS SUCCESSFUL)
  CONTINUE:
    SET VAR $FLAG EQ DONE
/*
```

SET WAIT—Set a Wait Interval

To delay processing between ZEKESET statements, specify the number of seconds (from 1 through 9999) to wait in the SET WAIT statement. When the number of seconds is passed, processing continues with the next statement. For example, ZEKESET waits at this statement for 10 seconds:

```
SET WAIT 10
```

You can use SET WAIT to allow a previous command time to process before issuing another command.

OPTION—Set ZEKESET Execution Options

Use the ZEKESET statement OPTION to specify these ZEKESET execution options:

Parameter	Description
NOSUB	Indicates to disable variable substitution during the execution of ZEKESET. This option is particularly useful if you have event names that do not begin with letters (e.g., names that begin with $). To re-enable variable substitution, you can specify OPTION SUB; or, simply exit the ZEKESET program. **Note:** The difference between using OPTION NOSUB and ZEKE-CTL NOSUB to turn off variable substitution is that OPTION NOSUB turns it off at statement execution time, while ZEKE-CTL NOSUB turns it off at variable substitution time (just prior to dispatch of the event). See the *ASG-Zeke Scheduling for z/OS User's Guide* for more information on ZEKE-CTL NOSUB.
SUB	Indicates to re-enable variable substitution.
NOZCOMPRINT	Indicates to reset writing Zeke command output back to the console (instead of to SYSPRINT) for executions of SET ZCOM.
ZCOMPRINT	Indicates to enable writing Zeke command output to SYSPRINT (instead of to the console) for subsequent executions of SET ZCOM. This option is particularly useful for commands that are likely to produce a large amount of output. To reset the writing of output back to the console, you can specify OPTION NOZCOMPRINT; or, simply exit the ZEKESET program.

Note:

When setting multiple options, you must specify each option in a separate OPTION statement. For example:

```
OPTION NOSUB
OPTION ZCOMPRINT
```

You cannot combine them on a single line. For example, this statement is invalid:

```
OPTION NOSUB ZCOMPRINT
```

Examples

This example executes the ZEKESET program without variable substitution:

```
//ZSET     JOB   ,ZEKESET.SAMPLE,CLASS=A
//ZSAMPLE  EXEC  PGM=ZEKESET,PARM='SUBSYS=ZDEV'
//SYSPRINT DD    SYSOUT=A
//SYSIN    DD    *
  OPTION NOSUB
  SET WAIT $VAR1
/*
```

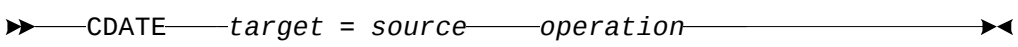
This example enables variable substitution in the ZEKESET program:

```
//ZSET     JOB   ,ZEKESET.SAMPLE,CLASS=A
//ZSAMPLE  EXEC  PGM=ZEKESET,PARM='SUBSYS=ZDEV'
//SYSPRINT DD    SYSOUT=A
//SYSIN    DD    *
  OPTION SUB
  SET WAIT $VAR1
/*
```

CDATE—Perform Date Calculations

The CDATE function of ZEKESET performs date calculations using Zeke variables. These variables can then be used for variable substitution. The variables are specified without the $, so that if Zeke submits the ZEKESET job, it does not perform variable substitution and overlay the variable names.

Syntax



Parameters

You can use these parameters with the CDATE statement:

Element	Description
target	The variable where the result is stored.

Element	Description
= (equal)	**Note:** You also can specify this element as `EQ`.
source	The variable that contains the date to be modified. Only one source variable is allowed. Dates cannot be added together.

operation	You can use an operator and modifier together, use the DAY.*x* parameter in addition to the operator and modifier, or use the DAY.*x* parameter alone:
	operator — A plus sign (+) or minus sign (-). This indicates whether the value is to be added to or subtracted from the numeric value specified.
	modifier — A numeric value (either a variable or an actual number) followed by one of these modifiers:
	`DAY` / `DAYS` — If only a numeral is specified, DAY(S) is the default.
	`WEEK` / `WEEKS` — Number of weeks.
	`MONTH` / `MONTHS` — Number of months.
	`YEAR` / `YEARS` — Number of years.
	`DAY.`*x* — CDATE supports the DAY.*x* and DAY.L keywords (same as OCCURS clauses). The DAY.*x* parameter determines the specified day of the month in question, where *x* is either a numeric value ranging from `1` through `31`, or the letter `L` (to determine the last day of the month).
	You can use the DAY.*x* parameter in addition to other operators and modifiers, or in place of an operator and modifier.
	Examples:
	If the value of INDATE is 12/14/2020, the eleventh day of the month is determined, and a final date value of 12/11/2020 is returned:
	`CDATE OUTDATE EQ INDATE DAY.11`
	If the value of INDATE is 12/14/2020, the last day of the month is determined, and a final date value of 12/31/2020 is returned:
	`CDATE OUTDATE EQ INDATE DAY.L`

Element	Description

	If the value of INDATE is 12/14/2020, one month is added to it, and the fifth day of the resulting month is determined. This yields a final date value of 01/05/2020:
	`CDATE OUTDATE EQ INDATE + 1 MONTH DAY.5`
	
	If the value of INDATE is 12/14/2020, one week is added to it, and the last day of the resulting month is determined. This yields a final date value of 12/31/2020:
	`CDATE OUTDATE EQ INDATE + 1 WEEK DAY.L`
	
	If the value of INDATE is 12/14/2020, the last day of the month is determined, and one week is added to the result. This yields a final date value of 01/07/2020:
	`CDATE OUTDATE EQ INDATE DAY.L + 1 WEEK`

Source/Target Variables

This is the default date format:

mm/dd/yyyy

Or

dd/mm/yyyy (if your system is set up for using European format).

To override this format, specify your preferred format in parentheses following the variable name. For example:

`VAR1(yyyyddd)`

If you attempt to perform a CDATE operation on two variables with different year formats (e.g., one in *yy* format and one in *yyyy* format), an error occurs.

The special name DATE can be used as a source variable and does not need any format specification. The value is the current system date. If you have a source variable $DATE, use the word VAR in front of it to distinguish it from the special name DATE.

If a proper format is not specified, or if a date cannot be determined, a syntax error results and the action taken depends on the ON ERROR setting. See "ON ERROR—Act on Syntax Errors" on page 195 for more information. See also "ON EOM—End-of-Month Rounding" on page 194 for tips on avoiding syntax errors due to invalid end-of-month date calculations.

If the source variable and target variable have the same name and format, omit the target variable and the equal sign. The result is stored into the source variable.

All CDATE input must be valid and non-conflicting.

Use these character strings in any combination to specify the format of a target or source variable:

String	Description
YYYY	Represents a 4-digit year.
YY	Represents a 2-digit year. If *YY* is greater than or equal to 50, it is considered a twentieth century date (19*xx*); if it is less than 50, it is considered a twenty-first century date (20*xx*).
MM	Represents a 2-digit month that ranges from 01 (January) through 12 (December).
MONA	Represents a 3-character month abbreviation (e.g., JAN, FEB, MAR, etc.).
MON	Represents a month name (e.g., JANUARY, FEBRUARY, etc.).
DAYN	Represents a numeric day that ranges from 1 (Monday) through 7 (Sunday).
DAYA	Represents a 3-character day abbreviation (e.g., MON, TUE, WED, etc.).
DAY	Represents a day name (e.g., MONDAY, TUESDAY, etc.).
DDD	3-digit Julian date of year that ranges from 001 through 366.
DD	Represents a 2-digit day of month that ranges from 01 through 31.

The user-specified format is scanned from left to right. The format elements are matched in the order in which they appear in the table (e.g., DDD is matched before DD). For example, if a format contains the string DDDDD, this string is interpreted as the 3-digit day of the year followed by the 2-digit day of the month.

Except for the DAYN format, the CDATE function always stores an alphanumeric value in the target variable so that leading zeros are preserved.

The size and value of a source variable are expected to be consistent with the source format specified; otherwise, an error results.

If an incomplete date format is specified, only modifiers relevant to the format are valid.

Examples:

In this statement, the only valid modifiers are MONTHS and YEARS. The DAYS modifier is invalid:

```
VAR1(YYYYMM)
```

In this statement, the MONTHS modifier is valid because even though it is not named in the format, it is relevant to the format:

```
VAR2(YYYYDDD)
```

If the source variable is formatted for only a day (DAY, DAYA, or DAYN), only DAY modifiers are allowed.

For example, if VAR1 contains the word TUESDAY, then this statement stores WEDNESDAY into VAR1:

```
CDATE VAR1(DAY) + 8 DAYS
```

If a variable contains both a complete date and a day value (e.g., `VAR1(DAY - mm/dd/yyyy`), then the complete date is used for all calculations and any DAY values come from the result of the complete date calculation.

For example, if VAR1 contains MONDAY - 01/21/2020, then this statement stores FRIDAY into VAR2, not WEDNESDAY:

```
CDATE VAR2(DAY) = VAR1(DAY - mm/dd/yyyy) + 2 DAYS + 1 YEAR
```

If a string in the format does not match any of the elements listed above, it is considered a literal. A literal in the format for a source variable is expected to have an exact match in that position in the source variable value; otherwise, an error results. A literal in a target variable format is simply copied to the target variable value at the position where it was encountered. This allows you to use characters such as spaces, dashes, slashes, commas, etc., to separate date elements in the source and target. You can also use numerical digits and alphabetical characters to manipulate the date value of the target.

Examples:

This statement shows a date separated by spaces:

```
VAR1(MM DD YYYY)
```

This statement shows a date separated by a slash:

```
VAR2(YYYY/DDD)
```

```
VAR3(MM-DD-YYYY)
```

This statement shows a date separated by dashes:

```
VAR4(MON DD,YYYY)
```

This statement shows a combination of spaces and commas:

```
VAR5(MM/01/YYYY)
```

This statement shows a numerical manipulation of date:

```
VAR6(DDD20YYYY)
```

You can set a variable value to a literal, then use the literal as input to subsequent CDATE processes to alter date formats. The initial variable's literal value must use a valid date format and be a valid date. You must specify the date (i.e., YYYYMMDD) after the source variable in the CDATE line.

For example, if you set a variable named $DATEFORCE to a literal value of 20200819 (the date value in YYYYMMDD format), the value can then be parsed in a subsequent CDATE operation to a different value, then parsed again in another, and so on, as in this example:

```
SET VAR $DATEFORCE TO '20200819'
CDATE DOUG1(DDMMYY) = DATEFORCE(YYYYMMDD)
CDATE DOUG2(YYMMDD) = DATEFORCE(YYYYMMDD)
CDATE DOUG3(YYYY) = DATEFORCE(YYYYMMDD)2020
```

Tips On Using CDATE

- CDATE statements can have conditional IF clauses similar to the IF clause of the SET statement. When the IF clause is true, Zeke processes the CDATE statement. When the IF clause is false, the CDATE statement is bypassed.
- All date operations are performed from left to right within the CDATE statement.
- Treat all CDATE operations output as alphanumeric data for comparison purposes.

Valid CDATE Examples

These are valid examples of the CDATE statement.

This statement adds one day to the current system date and the result is stored into the variable VAR1 in default format:

```
CDATE VAR1 = DATE + 1 DAY
```

In this statement, no format specification is needed for the special name DATE:

```
CDATE VAR2(YYYYDDD) = DATE + 3 WEEKS
```

In this statement, format specification is needed for VAR2 and VAR3 because the format is not the default format:

```
CD VAR3(MM-YYYY) = VAR2(YYYYDDD) - 5 DAYS
```

In this statement, DAY is assumed for the numeral 1. No format specification is needed for VAR1 because it is in the default format:

```
CD VAR1 EQ VAR1 + 1
```

Since the source and target variable names and formats are the same, the statement could be simplified to `CD VAR1 + 1`.

In this statement, VAR is needed because the source variable name is the same as the special name DATE:

```
CDATE VAR DATE = DATE + 3
```

In this statement, more than one set of operators and modifiers can be used:

```
CD VAR3(MON,DD,YYYY) EQ VAR2 + 3 DAYS + 2 MONTHS
```

On the last day of the month) this statement calculates the date of the last day of the previous month:

```
CD VAR1 = DATE + 1 DAY - 1 MONTH - 1 DAY
```

On the last day of the month, this statement calculates the date of the last day of the following month:

```
CD VAR2 = DATE + 1 DAY + 1 MONTH - 1 DAY
```

On the last day of the month, this statement calculates the date of the first day of the current month:

```
CDATE VAR4 = DATE + 1 DAY - 1 MONTH
```

On the last day of the month, this statement calculates the date of the first day of the previous month:

```
CDATE VAR5 = DATE + 1 DAY - 2 MONTH
```

Sample JCL

```
//STEP1    EXEC PGM=ZEKESET,PARM='SUBSYS=SSSI'
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *
  CDATE   CURRMNTH(MM) EQ DATE
  SET VAR $FISCALMTH EQ '12/04/2020' IF $CURRMNTH EQ '12'
/*
```

Invalid CDATE Examples

These are *invalid* examples of the CDATE statement.

This statement is invalid because the source variable must be the first word following the equals:

```
CDATE VAR1 = 3 DAYS + VAR2
```

This statement is invalid because Zeke cannot add two variables:

```
CDATE VAR1 = VAR2 + VAR3
```

This statement is invalid because the result cannot be stored into the special name DATE:

```
CDATE DATE + 3
```

If DATE is a variable name, code the statement:

```
CDATE VAR DATE + 3
```

This statement is invalid because VAR1 does not contain a DAY value:

```
CDATE VAR1(YYYYMM) + 3 DAYS
```

This statement is invalid because there is no way to determine whether it is a leap year:

```
CDATE VAR1(MMDD) + 20 DAYS
```

This statement is invalid because there is no way to determine when the next month is reached:

```
CDATE VAR1(DD) + 10 DAYS
```

This statement is invalid due to conflicting input values. If INDATE has a value of `4 FRIDAY`, an error results because day 4 of the week is Thursday, not Friday:

```
CDATE OUTDATE EQ INDATE(DAYN DAY)
```

This statement is invalid due to conflicting input values. If INDATE has a value of 2020001 - 01022020, an error results because day 001 of the year is 01012020, not 01022020:

```
CDATE OUTDATE EQ INDATE(YYYYDDD - MMDDYYYY)
```

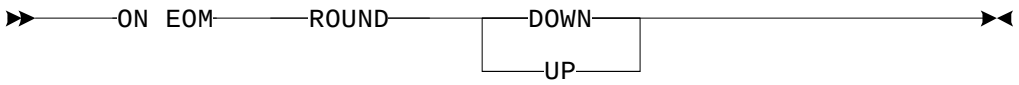
ON EOM—End-of-Month Rounding

The ON EOM statement is used to specify how you want ZEKESET to deal with date calculations in which the input or output date occurs at the end of a month. It establishes the rounding settings when it is invoked and those settings are in effect until another ON EOM statement is encountered.

Note:

The ON EOM and CDATE statements must be entered on separate lines.

Syntax



Parameters

You can use these parameters with the ON EOM statement:

Parameter	Description

DOWN	Output date is rounded down one day at a time until a valid date is reached. Use this keyword when the source month in a date calculation has more days than the target month.
	For example, if the source date is 03/31/2020 and this calculation is performed:
	`CDATE TARGDATE EQ SRCDATE + 1 MONTH`
	then, the initial result would be 04/31/2020 (which is an invalid date).
	By specifying ROUND DOWN:
	`ON EOM ROUND DOWN`
	`CDATE TARGDATE EQ SRCDATE + 1 MONTH`
	one day is subtracted from the result until a valid date is reached. The result is the last day of the target month (i.e., 04/30/2020).

ROUND	Activates UP and DOWN rounding.
	`ON EOM ROUND DOWN`
	`ON EOM ROUND UP`

Parameter	Description

UP	Output date is rounded up one day at a time until the last valid date of that month is reached. Use this keyword when the source month used in a date calculation has fewer days than the target month.
	For example, if the source date is 02/28/2020 and this calculation is performed:
	`CDATE TARGDATE EQ SRCDATE + 1 MONTH`
	then, the initial result would be 03/28/2020 (which is not the last day in March).
	By specifying ROUND UP keyword:
	`ON EOM ROUND UP`
	`CDATE TARGDATE EQ SRCDATE + 1 MONTH`
	one day is added to the result until the last date of the target month is reached (i.e., 03/31/2020).

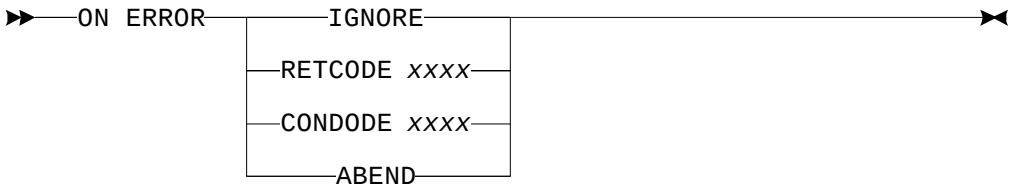
ON ERROR—Act on Syntax Errors

The ON ERROR statement specifies the action to take when syntax errors are found in the input control statements supplied to ZEKESET.

The default is to immediately exit the ZEKESET program with a condition code of 4, 8, or 12 depending on the type of error. The other options are to ignore the error or to set a specified condition code.

The ON ERROR statement can be used more than once. The last encountered ON ERROR statement is the one used if a syntax error is found.

Syntax



Parameters

You can use these parameters with the ON ERROR statement:

Parameter	Description
`IGNORE`	Ignores the syntax errors and continues with the next ZEKESET input statement.
`RETCODE` *xxxx*	Exits the ZEKESET program immediately with the specified return code (up to four digits long ranging from `0` through `4096`).
`CONDCODE` *xxxx*	Exits the ZEKESET program immediately with the specified condition code (up to four digits long ranging from `0` through `4096`).
`ABEND`	Exits the ZEKESET program immediately with user abend code U0001.

Sample JCL

The ON ERROR IGNORE statement in this example ignores the syntax errors and continues with the next ZEKESET input statement:

```
//JOBNAME  JOB ...
//STEP01   EXEC PGM=ZEKESET,PARM='SUBSYS=SSSI'
//STEPLIB  DD   DSN=ZEKE.LINKLIB,DISP=SHR
//         DD   DSN=OASIS.LINKLIB,DISP=SHR
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *
  ON ERROR IGNORE
  SET ZCOM 'XXX'
  SET ZCOM 'ZD'
/*
```

The ON ERROR statement in this example exits the ZEKESET program immediately with the return code 020.

```
//TLAZSETR JOB (10039),'NAME',NOTIFY=TSTLAA,
//          MSGCLASS=X,MSGLEVEL=(1,1),CLASS=A
//STEP01   EXEC PGM=ZEKESET,PARM='SUBSYS=SSSI'
//STEPLIB  DD   DSN=ZEKE.LINKLIB,DISP=SHR
//         DD   DSN=OASIS.LINKLIB,DISP=SHR
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *
  ON ERROR RETCODE 020
  SET ZCOM 'XXX'
  SET ZCOM 'ZD'
/*
```

The ON ERROR statement in this example exits the ZEKESET program immediately with the return code U0001:

```
//TLAZSETA JOB (10039),'NAME',NOTIFY=TSTLAA,
//          MSGCLASS=X,MSGLEVEL=(1,1),CLASS=A
//STEP01   EXEC PGM=ZEKESET,PARM='SUBSYS=SSSI'
//STEPLIB  DD   DSN=ZEKE.LINKLIB,DISP=SHR
//         DD   DSN=OASIS.LINKLIB,DISP=SHR
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *
  ON ERROR ABEND
  SET ZCOM 'XXX'
  SET ZCOM 'ZD'
/*
```

The ON ERROR statement in this example exits the ZEKESET program immediately with the condition code of 018:

```
//TLAZSETC JOB (10039),'NAME',NOTIFY=TSTLAA,
//          MSGCLASS=X,MSGLEVEL=(1,1),CLASS=A
//STEP01   EXEC PGM=ZEKESET,PARM='SUBSYS=SSSI'
//STEPLIB  DD   DSN=ZEKE.LINKLIB,DISP=SHR
//         DD   DSN=OASIS.LINKLIB,DISP=SHR
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *
  ON ERROR CONDCODE 018
  SET ZCOM 'XXX'
  SET ZCOM 'ZD'
/*
```

4 Report Writer

The Report Writer facility is a function of the ZEKE batch utility program. This chapter describes the types of reports, the parameters for record selection, how to format the reports, and how to execute the facility. It discusses these topics:

Topic	Page
Generating Reports	200
Output Reports to File	203
Report Formatting	204
Report Sequence—Event/Plan Listings	204
Page Breaks—Event/Plan Listings	205
Line Spacing—Event/Plan Listings	206
Report Output—Event/Plan Listings	207
Selecting Fields—Event/Plan Listings	208
LIST EVENTS/PLAN—EMR/SQR Listings	213
LIST CALID—Zeke Calendar Information	234
LIST CREDENTIAL—Zeke Credentials	237
LIST GENOPTS—Zeke Generation Options	239
LIST LOGIN—Zeke Logins	241
LIST OPERCLAS—Operator ID Security Classes	243
LIST OPERRECS—Zeke Operator ID Information	245
LIST RESOURCE—System Resources	247
LIST VARIABLE—Zeke Variable Information	249

Generating Reports

The LIST command generates Zeke reports. Many parameters are available to be used with the LIST command to generate each report type. Only a few samples are shown in this procedure.

Security authorization is performed to ensure the user requesting the report has at least READ access to the requested records. To generate a LIST PLAN report, the user must be authorized for at least READ access to both the EMR and the SQR.

To generate a report

1 Create a jobstream that executes the ZEKEUTL procedure and include the desired LIST commands. The ZEKEUTL procedure executes the Zeke program. This is a sample jobstream using the ZEKEUTL procedure:

```
//ZUTIL  JOB , MSGLEVEL=(1,1),CLASS=A
//Z1     EXEC ZEKEUTL,PARM='SUBSYS=SSSI'
//SYSIN  DD *
  LIST EVENT . . . .
  LIST PLAN . . . .
  LIST VAR . . . .
  LIST CALID . . . .
  LIST RESOURCE . . . .
  LIST OPERCLAS . . . .
  LIST OPERRECS . . . .
```

2
```
/*
/&
* $$ EOJ
```
Use these LIST commands (depending on the desired result):

Command	Description

Produce an event listing	`LIST EVENT [`*`selection-parameters`*`]`
	`LIST EVENTS [`*`selection-parameters`*`]`
	For valid selection parameters, see "LIST EVENTS/PLAN—EMR/SQR Listings" on page 213.

Produce a scheduled event listing	`LIST PLAN [`*`selection-parameters`*`]` For valid selection parameters, see "LIST EVENTS/PLAN—EMR/SQR Listings" on page 213.
Produce a calendar listing	`LIST CALID [`*`selection-parameters`*`]` For valid selection parameters, see "LIST CALID—Zeke Calendar Information" on page 234.

Command	Description
Product a Zeke credential listing	`LIST CRED` (This command has no selection parameters.)
Produce a GENOPT listing	`LIST GENOPTS` (This command has no selection parameters.)
Product a Zeke login listing	`LIST LOGIN` (This command has no selection parameters.)
Produce a security class listing	`LIST OPERCLAS` (This command has no selection parameters.)
Produce an operator ID listing	`LIST OPERRECS` (This command has no selection parameters.)
Produce a resource listing	`LIST RESOURCE` (This command has no selection parameters.)
Produce a variable listing	`LIST VAR [`*`selection-parameters`*`]` For valid selection parameters, see "LIST VARIABLE—Zeke Variable Information" on page 249.

3 To add a report title to the top of the report, use the TITLE parameter followed by name of the report. If the title contains more than one word, it must be enclosed in delimiters (e.g., quotes or parentheses). The title can be up to 60 characters, and the keyword TITLE can be abbreviated TITL.

For example, to produce a report with the title SCHEDULE LISTING FOR SYSTEM X BY USERID, use this command:

```
LIST PLAN TITLE 'SCHEDULE LISTING FOR SYSTEM X BY USERID'
```

4 To change the line spacing from the default of single-spacing to alternative line spacing, use one of these parameters.

- To double-space between entries, use SPACE2.
- To triple-space between entries, use SPACE3.

For example, to produce a double-spaced report for all events, use this command:

```
LIST EVENT SPACE2
```

5 To specify the sort sequence for the report, use the SEQUENCE parameter followed by the desired keywords contained in a single set of parentheses. Valid keywords are listed in "Report Sequence—Event/Plan Listings" on page 204

For example, to select SYSTEM A events and sequence by event number within event type, use this command:

```
LIST EVENT SYS A SEQUENCE (TYPE)
```

6 To add page breaks that sort by sequence, specify PAGEBREAK with the same SEQUENCE parameter followed by the desired keywords contained in a single set of parentheses. Valid page break keywords are listed in "Page Breaks—Event/Plan Listings" on page 205.

For example, to produce an Event Listing of all events, sorted by event type within the same system ID, and display every new event type on a new page, use this command:

```
LIST EVENT SEQUENCE (SYSTEM TYPE) PAGE (TYPE)
```

7 To specify fields to be printed on the report, use the FIELDS parameter followed by the desired keywords contained in a single set of parentheses. Valid FIELDS keywords are listed in "Selecting Fields—Event/Plan Listings" on page 208.

For example, to product an Event Listing of all events containing the event description field and last date of last dispatch field, use this command:

```
LIST EVENT FIELDS (DESC DISPDATE)
```

8 To use a data space when generating the report, include the DATASPACE keyword. The DATASPACE keyword is valid for all Report Writer jobs. Using a data space allows the Report Writer program to execute more quickly.

Note:

If you always want to use a data space for generating reports, set the DSPBatch generation option to Y (see page 541). If you do so, all Report Writer jobs will use a data space to generate a report, regardless of whether the DATASPACE parameter is specified. (To override this setting temporarily, you can specify the NODATASPACE parameter.)

```
LIST EVENTS DATASPACE
LIST PLAN DATASPACE
```

Note:

If you run multiple LIST statements in the same execution of the ZEKE utility program, all statements will use the same data space (which eliminates the overhead of creating multiple data spaces).

9 Once you have completed your jobstream, execute the ZEKEUTL program to generate the report.

Output Reports to File

Use the OUTFILE parameter to write selected records to a work file for user processing. The output file can be written to DASD or tape. The OUTFILE keyword can be followed by an operand naming the ddname for the output file. The default ddname is ZEKEWK.

When the OUTFILE parameter is used, a DD statement for the output dataset (ZEKEWK, or the user-specified name) must be available in the jobstream. Each LIST command opens the dataset, writes selected records, and then closes the dataset. If multiple LIST commands refer to the same DD, only the first report is written to the dataset.

The record layout of the output record is in Zeke install library member ZEKERCD. The record consists of three parts:

- The schedule queue record extract—not used for an Event Master Listing
- The WHEN condition for the schedule queue record (LIST PLAN) or the lowest version WHEN condition (LIST EVENT)
- The EMR extract

This is the output dataset DCB information:

```
RECFM = VB
LRECL = 4000
BLKSIZE = 4008
```

To create a sequential disk file

1. Create a jobstream using the LIST command with the OUTFILE parameter to write selected records to a file. For example, to write a scheduled event named JOB200 to a file, use this command:

   ```
   LIST PLAN JOB JOB200 OUTFILE
   ```

2. To name the output file, specify the ddname in a single set of parentheses after the OUTFILE parameter. For example, to write scheduled event JOB200 to an output file called USRNAME, use this command:

   ```
   LIST PLAN JOB JOB200 OUTFILE (USRNAME)
   ```

3. To override the DFSORT message from appearing in Zeke report output, use these DD card and SYSIN parameters:

   ```
   //DFSPARM DD *
   OPTION MSGDDN=SYSOUT
   ```

4. After you complete your jobstream, execute the ZEKEUTL procedure to generate the file.

Report Formatting

Several parameters are available to format many of the report types. You can select the sequence of the information, control the page breaks and spacing, select additional information and the fields to be printed, and choose the output format (i.e., print or file).

Report Sequence—Event/Plan Listings

The SEQUENCE parameter specifies the sort sequence for the Event and Plan Listings. Default sequence is event number for LIST EVENT(S) and schedule time for LIST PLAN.

Note:

You can specify the EVENT function either as `EVENT` or `EVENTS`.

Parameters

You can specify the SEQUENCE parameter followed by up to five keywords contained within a single set of parentheses. Specify major sort fields first.

Parameter	Description	
`SEQuence`	Indicates the sort sequence for the report. These are the valid keywords:	
	`APPLICAT`	Sort by application ID.
	`CALID`	Sort by calendar ID.
	`DISPDATE`	Sort by last dispatch date.
	`DISPPRI`	Sort by dispatching priority.
	`DISPTIME`	Sort by last dispatch time.
	`EARLY`	Sort by early time.
	`ENAME`	Sort by event name.
	`EVENT`	Sort by event number.
	`GROupid`	Sort by group ID.
	`HITDATE`	Sort by date the event was selected.
	`JOBNAME`	Sort by jobname.
	`LATEstart`	Sort by 'late start' time.

Parameter	Description	
	LATEEND	Sort by 'late end' time.
	LOGINName	Sort by Zeke login name.
	SCHED	Sort by schedule time.
	SCHENV	By scheduling environment name
	SECGroup	Sort by security group (z/OS only).
	SELDATE	Sort by date schedule record was built.
	STATtime	Sort by time of last event status change (e.g., dispatch, BOJ, EOJ, etc.).
	SYSTEM	Sort by owning system ID.
	TASKTYPE	Sort by task event type (e.g., sFTP).
	TRIGger	Sort by the value of the event trigger option.
	TYPE	Sort by event type.
	USERid	Sort by user ID.
	VARiable	Sort by variable name.
	VERsion	Valid with LIST PLAN only. Sort by version number.

Page Breaks—Event/Plan Listings

The PAGEBREAK parameter specifies the field that controls page breaks on the report. Every time the value of this field changes, a new page is created. To avoid an abundance of page breaks, specify a PAGEBREAK field that also is a field specified as a SEQUENCE parameter.

Parameters

You can specify the PAGEBREAK parameter followed by keywords contained within a single set of parentheses.

Parameter	Description	
PAGEbreak	Indicates the field that controls page breaks on the report. These are the valid keywords:	
	APPLICAT	Application ID.
	CALID	Calendar ID.
	DISPPR	Dispatching priority.
	EVENT	Event number.
	GROUPID	Group ID.
	HITDATE	Due date.
	LOGINName	Zeke login name.
	SELDATE	Select date.
	SYSTEM	System ID.
	TASKTYPE	Task event type (e.g., sFTP).
	TYPE	Event type.
	USERID	User ID.
	For example, this command produces an EMR listing of all events, sorted by event type within each system ID: `LIST EVENTS SEQUENCE (SYSTEM TYPE) PAGE (TYPE)` Every new event type is printed on a new page.	

Line Spacing—Event/Plan Listings

The reports are single-spaced. You can use these parameters to specify alternate spacing options:

Parameter	Description
SPACE2	Print the report double-spaced.

Parameter	Description
SPACE3	Print the report triple-spaced.

Report Output—Event/Plan Listings

The FORMAT parameter specifies the report output format (i.e., formatted report file or a comma-separated values (CSV) file).

The FORMAT keyword is available only for LIST EVENT or LIST PLAN. Output for other report types is written to SYSPRINT.

Parameter

This parameter is used to indicate the report format:

Parameter	Description	
FORMAT	Specifies the output format for the report. These are the valid keywords:	
	PRINT	Default. Requests a printed report. If a report is written to a dataset, the record size of the dataset must be long enough to report all fields specified. If the length is too short, the report is not printed.
	CSV	Requests a comma-separated values (CSV) file to be written to the CSVOUT DD statement. **Note:** Only fields can be included in CSV output (see "Selecting Fields—Event/Plan Listings" on page 208); other types of non-field keywords (e.g., LISTWHEN) are ignored. The first line in the file contains a header line. No page breaks are generated; if you specify the PAGEBREAK keyword, it is ignored. The CSVOUT data set will be overwritten as: `LRECL=32760, BLKSIZE=32764, RECFM=VB`

Selecting Fields—Event/Plan Listings

The FIELDS parameter specifies the fields to be printed on the report.

Parameters

You can specify the FIELDS parameter followed by keywords contained within a single set of parentheses. Each keyword represents a field on the event master or event schedule records, or both. The record used depends on the LIST command (PLAN=SQRs; EVENTS=EMRs).

For a report directed to a dataset, use the LRECL of the dataset to determine the length of the report line. When calculating line length, add one character between fields.

Parameter	Description
FIELDs	Specifies the fields to be printed on the report. Separate the keywords with a comma, and enclose all of the keywords in parentheses. `LIST EVENTS FIELD=(APP,ENAME,PLAT,SYS,SCHED)` **Note:** Each output field that you can display on the report has a maximum length. For example, if you choose to print the event description (i.e., using the DESCRIPT keyword) on the report, then the output for that field can be up to 60 bytes long. These are the valid keywords (including the maximum length of the output for each keyword): **Keyword** — **Length** — **Description** APPlicat — 8 — Application ID. CALid — 8 — (From the EMR) Event calendar ID. CANcelco — 5 — (From the SQR for job events only; valid for LIST PLAN only) Last cancel/abend/flush code. CONTrol — 3 — Tracking indicator for Zeke-controlled jobs. DESCript — 60 — (From the EMR) Event description. DISPCount — 6 — (From the EMR) Number of dispatch times. DISPDate — 10 — Last dispatch date. DISPPrio — 3 — Dispatching priority.

Parameter	Description		
	DISPTime	8	Time of last dispatch (in *HH:MM:SS* format).
	DRL	3	Disaster recovery level.
	EARLytim	5	Early time.
	ENAMe	12	Event name.
	EVent	6	Event number.
	EXCeptio	20	(From the SQR; valid for LIST PLAN only) Exception processing reasons.
	EXPire	10	Event expiration date.
	FREQCalc	3	Frequency calculation for recurring events based on schedule time or clock time.
	FREQUenc	5	Dispatch frequency for recurring events.
	GROupid	3	Group ID.
	HITdate	10	(From the SQR; valid for LIST PLAN only) Schedule date.
	JOBname	8	Jobname.
	LATEEND	5	'Late end' time. (See "LATEEND" on page 39.)
	LATEstart	5	'Late start' time. (See "LATEstart" on page 39.)
	LOGINName	32	Zeke login record assigned to the task event.
	MILESTone	3	YES or NO to indicate the value of the MILESTONE parameter. (See "MILESTone" on page 40.)
	MULTihit	3	Allow multiple SQRs because of a nonworking day.
	MUSTend	5	Time by which event must end. (See "MUSTend" on page 40.)
	NOTAfter	5	Time after which Zeke does not dispatch the event. (See "NOTAFter" on page 40.)
	NWDAY	3	How to schedule on nonworking days.

Parameter	Description		
	OPErok	3	Value of the OPEROK parameter. Valid values are YES and NO.
	PERManent	3	Value of the PERM parameter. (See "PERManent" on page 41.) Valid values are YES and NO.
	PLATform	8	Operating system where the event will execute.
	RETain	3	Value of the RETAIN parameter. (See "RETAin" on page 42.) Valid values are YES and NO.
	SCHEdtim	5	Schedule time.
	SCHENV	16	Scheduling environment name
	SELdate	10	(From the SQR; valid for LIST PLAN only) Date the SQR was created.
	STATTime	8	(From the SQR; valid for LIST PLAN only) Time of last event status change (e.g., dispatch, BOJ, EOJ, etc.).
	STATUS	9	(From the SQR; valid for LIST PLAN only) SQR status (e.g., HOLD, PEND, FAIL, etc.).
	SYStem	8	System ID that owns event.
	TASKTYPE	8	Task event type (e.g., sFTP)
	TEXt	62	Event description. These are the descriptions depending on the event type: • Command text for *x*COM events • REXX exec name and arguments for REXX events • Operator message lines for message events • WorkCtr lines for work centers • First description line text for job events.

Parameter	Description		
	TEXT1	62	First line of the event description. These are the descriptions depending on the event type: • Command text for *x*COM events • REXX exec name and arguments for REXX events • Operator message lines for message events • WorkCtr lines for work centers • First description line text for job events.
	TIMes	4	Number of dispatch times (up to 9999).
	TRIGger	3	Event trigger option(s).
	TYPE	4	Event type.
	USErid	8	Event user ID.
	VERLoad	5	(From the EMR) Version load count (i.e., number of versions of this event to be loaded during a schedule build).
	VERsion	5	(From the SQR; valid for LIST PLAN only) Event version number.
	These keywords specify fields to print when reporting on job events:		
	AUTOReply	3	Indicates the value of the AUTO REPLIES parameter. Valid values are YES and NO.
	AVGDur	8	(From the EMR) Average duration.
	CANCELCO	5	Valid only for job events for LIST PLAN. Condition code that caused the job or step to fail or cancel.
	CLAss	6	Class list
	JCLSOurc	17	(From the EMR) JCL source (i.e., file type, DD or library name, and member name).
	Jobname	8	Jobname.
	LJOBname	30	Long jobname (i.e., longer than eight characters, or mixed-case of any length).

Parameter	Description		
	LJCLsrc	37	Long JCL source (i.e. file type, DD or library name, and member name).
	PRIority	3	Job priority.
	SECGrp	8	(z/OS only.) Security group.
	TAPes	3	Number of required tapes.
	TARGet	8	Netregid of the execution system.

Generic Selection Criteria—Event/Plan Listings

You can request a report using wildcard or placeholder characters in most selection criteria. Wildcards and placeholders can be used in operands for these parameters:

- Application ID
- Jobname
- Event name
- Group ID
- Scheduling environment name
- Security group
- Event calendar
- User ID

You can use an asterisk (*) as a wildcard for any number of characters. A wildcard functions in these ways:

- An asterisk at the end of an operand string selects any name (of any valid length) that *begins* with the specified characters. For example, this command selects jobs with such event names as PAYR1JOBABC, PAYR22, PAYRXXJOB:

  ```
  LIST EVENTS ENAME PAYR* LISTACC
  ```

- An asterisk at the beginning of an operand string produces *no* results. For example:

  ```
  LIST EVENTS ENAME *PAYR LISTACC
  ```

- An asterisk in the middle of an operand string performs a wildcard search for any name matching the specified beginning and ending characters, plus any characters in between.

  For example, this command selects jobs with such event names as PAYJOB, PAY1JOB, PAY22JOB, PAYXXJOB:

```
LIST EVENTS ENAME PAY*JOB LISTACC
```

This command selects jobs with such event names as PAYJOB, PAY1JOB, PAY22JOB, PAYXXJOB, PAYXXJOBYY:

```
LIST EVENTS ENAME PAY*JOB* LISTACC
```

You can use a question mark (?) as a placeholder for any single character.

For example:

This command selects jobs with such event names as PAYR1JOB, PAYR2JOB, PAYR3JOB, PAYRXJOB:

```
LIST EVENTS ENAME PAYR?JOB LISTACC
```

This command selects any events with a name that has PAY in positions one through three, and 01 in positions five and six:

```
LIST PLAN ENAME PAY?01?
```

Wildcards and placeholders can be used in combination. For example, this command selects job events with jobnames beginning with CL and have a P in the fifth position of the jobname:

```
LIST EVENTS JOB CL??P*
```

LIST EVENTS/PLAN—EMR/SQR Listings

You use this command to list EMR or SQR information from the Zeke database. The EMR listing (EVENTS) and schedule listing (PLAN) use the same reporting parameters. The schedule listing has additional parameters that also are described in this section.

Note:

To generate a LIST PLAN report, the user must be authorized for at least read access to both the EMR and SQR.

Parameters

All parameters have an AND relationship except for the Event Type parameters (which have an OR relationship). If you specify the Event Type parameter, the event must match all criteria to be selected.

These are the valid selection and report generation parameters:

Parameter	Description
ABEnd	Valid for a schedule listing (LIST PLAN) only. Selects events that completed abnormally.
ACTive	If not specified for an EVENTS Listing, selects both active and deactivated events. For example, this commands selects active job events: `LIST EVENTS ACTIVE JOB`
ADDed	Valid for a schedule listing (LIST PLAN) only. Selects events that were added to the schedule with the ZADD command.
ALTEred	Valid for a schedule listing (LIST PLAN) only. Selects events that were altered by a Zeke command, such as ZALTER or ZENABLE.
ALTNOTDur	Valid for a schedule listing (LIST PLAN) only. Selects events for which the ZALTER command was used with the NOTDUROK parameter.
ALTTimeok	Valid for a schedule listing (LIST PLAN) only. Selects events for which the ZALTER command was used with the TIMEOK parameter.
ALTWhenok	Valid for a schedule listing (LIST PLAN) only. Selects events for which the ZALTER command was used with the WHENOK parameter.

APPL	Selects events with the specified application ID (up to eight characters long). You can use wildcard and placeholder characters in your selection criteria (see "Generic Selection Criteria—Event/Plan Listings" on page 212). For example:
	`LIST PLAN APPL ABC`
	Select scheduled events with an application ID of ABC.
	`LIST PLAN APPL T*`
	Select scheduled events with an application ID beginning with T.
	`LIST EVENT APPL A?C`
	Select events with an application ID beginning with A, ending with C, and with any character in the second position.
	`LIST EVENT APPL ABC*`
	Select events with an application ID beginning with ABC.
	Note:
	Use delimiters when values might be confused with other Zeke parameter names.

BACklog	Valid for a schedule listing (LIST PLAN) only. Selects the schedule records for the prior working day.

Parameter	Description

CLass	Valid for job events only. Select events that have the specified class (up to two characters long) in its class list. If left blank, all job events that have a class ID are selected.
	
	Examples:
	
	This command selects job events that have at least one class specified:
	
	`LIST EVENTS CLASS`
	
	This command selects scheduled job events that have class A or class X in their class list:
	
	`LIST PLAN CLASS (A,X)`
	
	Note:
	The class is the class of the *event*, not the JCL class.

COMpleted	Valid for a schedule listing (LIST PLAN) only. Selects events with a normal completion status (not a pending or abend status).

DATASPACE	Indicates to use a data space when generating the report. For example:
	
	`LIST EVENTS DATASPACE`
	
	`LIST PLAN DATASPACE`
	
	Note:
	If you always want to use a data space for generating reports, you can set the DSPBatch generation option to Y (see page 541). This specifies for all Report Writer jobs to use a data space by default to generate reports (regardless of whether this DATASPACE parameter is specified).

DEACtivate	Valid for an events listing (LIST EVENT) only, because all schedule records are active. Selects only deactivated events. If not specified for an EVENTS listing, selects both active and deactivated events. For example, this command lists all deactivated events: `LIST EVENTS DEAC`
DISabled	Valid for a schedule listing (LIST PLAN) only. Selects disabled events.
DOne	Valid for a schedule listing (LIST PLAN) only. Selects completed events, regardless of the status (normal, abnormal, or disabled).
EARlytime	Selects events with early dispatch times. For example, this command lists only events that have an early dispatch time: `LIST EVENTS EARLY`

Parameter	Description
ENAMe	Selects events with the specified event name (up to 12 characters long). You can use wildcard and placeholder characters in your selection criteria (see "Generic Selection Criteria—Event/Plan Listings" on page 212). For example, this command lists scheduled events with a name that has PAY in the first through third positions, and 01 in the fifth and sixth positions: `LIST PLAN ENAME PAY?01?`

EVENTs	When used without the PLAN parameter, this parameter requests an event listing based on EMRs.
	When used with the PLAN parameter, this parameter requests a schedule listing based on scheduled events. You can enter one or more event numbers and use the RANGE, EXCLUDE, and EXRANGE parameters to select specific events.
	Examples:
	This command produce a standard event report:
	`LIST EVENTS`
	This command reports on these events (a maximum of 31 event numbers can be specified):
	`LIST EVENTS (33,34,88,134,332)`
	This command plans a report on only these scheduled events:
	`LIST PLAN EVENTS (1,2,3,4,5)`

EVTCal	Selects events with the specified calendar ID (up to eight characters long). You can use wildcard and placeholder characters in your selection criteria (see "Generic Selection Criteria—Event/Plan Listings" on page 212).
	Examples:
	This command lists scheduled events with calendar ID ACCTGQ11:
	`LIST PLAN EVTCAL ACCTGQ11`
	This command selects events with a calendar ID beginning with ACCTG:
	`LIST EVENT EVTCAL ACCTG*`
	This command selects events with a calendar ID beginning with A, ending with 2, and with any character in positions 2 through 7:
	`LIST EVENT EVTCAL A*2`

EXCEptions	Valid for a schedule listing (LIST PLAN) only. Selects events with an exception recorded. For example, ZALTER of the schedule time, events run late, added, held, disabled, or refreshed.

Parameter	Description
EXCLude	Specifies the events to exclude from the report. Any event number listed is excluded, even if it falls within a selected range or it is specified with the EVENTS parameter. The maximum number of events that can be listed following the EXCLUDE parameter (without overlaying storage in the initiator) is 31. For example, this command reports on events 100 through 200 except for events 104, 107, and 108: `LIST EVENTS RANGE (100,200) EXCLUDE (104, 107, 108)`

EXRange	Specifies the range of events to exclude from the report. Enter the starting and ending event number and enclose in parentheses. More than one range can be entered. A starting number without an ending is assumed to go to the end of the database.
	Examples:
	This command reports on all events except events 10 through 20:
	`LIST EVENTS EXRANGE (10,20)`
	This command reports on events 100 through 200, except events 130 through 150 and 180 through 185:
	`LIST EVENTS RANGE (100,200) EXRANGE (130,150,180,185)`

GROupid	Selects events with the specified group ID (up to three characters long). You can use wildcard and placeholder characters in your selection criteria (see "Generic Selection Criteria—Event/Plan Listings" on page 212).
	Examples:
	This command lists scheduled events with a group ID of 001:
	`LIST PLAN GROUP 001`
	This command lists scheduled events with a group ID that begins with a B:
	`LIST PLAN GROUP B*`
	This command lists events with a group ID beginning with Q, ending with 2, and with any character in the second position:
	`LIST EVENTS GROUP Q?2`
	This command lists events with an group ID beginning with QP:
	`LIST EVENTS GROUP QP*`
	Note:
	Use delimiters when values might be confused with other Zeke parameters.

HOld	Valid for a schedule listing (LIST PLAN) only. Selects events that are currently on hold.

Parameter	Description

JOB	Selects job events that match the specified criteria. You can use wildcard and placeholder characters in your selection criteria (see "Generic Selection Criteria—Event/Plan Listings" on page 212).
	
	If entered alone, this selects all job events. Can be combined on a command line with other event type parameters. (All event types are selected if no event type is specified.)
	
	Examples:
	
	This command selects only job events:
	
	`LIST EVENTS JOB`
	
	This command selects job events with jobnames beginning with PR:
	
	`LIST EVENTS JOB (PR*)`
	
	This command selects job events with jobnames beginning with CL and that have a P in the fifth position of the jobname:
	
	`LIST EVENTS JOB CL??P*`

LAte	Valid for a schedule listing (LIST PLAN) only. Selects scheduled events that are flagged as late.
LATETime	Selects events that have a late start time specified, regardless of their LATE status. For example, this command selects all events on system C that have a late start time specified: `LIST EVENTS LATET SYS C`
LISTAcctg	Prints the accounting information for each event. This information includes the number of dispatches and date and time of the last dispatch. For example, this command produces an EVENT report with dispatch information: `LIST EVENTS LISTACCTG`
LISTCond	Prints the defined condition codes for each event. For example, this command produces an EVENT report with condition code information: `LIST EVENTS LISTCOND`
LISTDESc	Prints the event description on the first line for each event. The detailed event information is printed below the description. For example, this command produces an EVENT report with event descriptions: `LIST EVENTS LISTDESC`
LISTDOc	Prints all event documentation. This is the same as specifying LISTNOTE, LISTDSN, LISTTEXT, and LISTPAD.
LISTDSn	Prints the defined tape datasets and current volume serial numbers for each event.

Parameter	Description
LISTEXit	Invokes user exit ZEKE02MX once for each event after all other reporting for the event is done. See the *ASG-Zeke Scheduling z/OS Installation Guide* for more information on the ZEKE02MX exit.
LISTJcl	Prints the JCL defined in the database. The JCL follows the detailed event information.
LISTNotes	Prints the note information for each event. For example, this command produce an EVENT report with note information. `LIST EVENTS LISTNOTES`
LISTOccurs	Prints the OCCURS clause for each event. For example, this command produces an EVENT report with OCCURS clause information: `LIST EVENTS LISTOCCURS`
LISTPad	Prints the scratch pad information for each event and comment lines for work centers.
LISTReply	Prints the automatic replies and messages for each event.
LISTRESO	Prints the resource information for each event.
LISTTASK	Prints detailed information (e.g., host name, Zena FTP commands, etc.) for each task event.
LISTText	Prints the text information for each event.
LISTVARS	Prints the local variables (name and value) for each event.
LISTVols	Prints the required disk volumes following the detailed event information.
LISTWhen	Prints WHEN conditions for each selected event. For work centers, prints SET conditions.
LISTWHY	Valid for a schedule listing (LIST PLAN) only. Prints the event status reason for each selected event. `LIST PLAN LISTWHY`
LOGINName	Selects task events that use the specified Zeke login record. You can use wildcard and placeholder characters in your selection criteria (see "Generic Selection Criteria—Event/Plan Listings" on page 212). For example, this command selects task events that are assigned with the Zeke logins ABCTCOM and ABCSUPPORT: `LIST EVENTS TASK LISTTASK LOGINN ABC*`

Parameter	Description

MILESTone	Selects events based on whether they are defined as milestone events.
	Examples:
	This command selects only events that are defined as milestones:
	`LIST EVENT MILEST=YES`
	This command selects only events that are not defined as milestones:
	`LIST EVENT MILEST=NO`

MSG	Selects message events that match.
	If entered alone, this selects all message events. Can be combined on a command line with other event type parameters. (All event types are selected if no event type is specified.)
	Examples:
	This command selects message events:
	`LIST EVENTS MSG`
	This command selects message events and Zeke command events:
	`LIST EVENTS MSG ZCOM`

NODATASPACE	If the DSPBatch generation option is set to Y (see page 541), this parameters indicates that the ZEKE utility program uses a data space when generating reports. To override this for a particular execution so that a data space is not used, use the NODATASPACE parameter.
NOEARly	Selects events that do *not* have an early dispatch time. For example, this command selects only events with no early dispatch time: `LIST EVENTS NOEARLY`
NOLate	Select events that *do not* have a 'late start' time. For example, this command selects only events with no 'late start' time specified: `LIST EVENTS NOLATE`
NOLATEEND	Select events that *do not* have a 'late end' time. For example, this command selects only events with no 'late end' time specified: `LIST EVENTS NOLATEEND`
NOOperok	Selects events that do *not* require an operator OK prior to event dispatch. For example, this command selects job events that do not require an OPEROK: `LIST EVENTS JOB NOOPEROK`
NOTDone	Valid for a schedule listing (LIST PLAN) only. Selects schedule records that are not complete (not DONE or disabled).

Parameter	Description
OCCursdetail	Valid for an Event Listing (EVENT) only. Selects events with OCCURS clauses that contain the specified keywords. For example, this command lists events that have an OCCURS clause that contains the keyword MONDAY: `LIST EVENTS OCCURSDETAIL (MONDAY)`
OPErok	Selects events that require an operator OK prior to dispatch (i.e., OPEROK=YES). For example, this command lists job events that require OPEROK: `LIST EVENTS JOB OPEROK`
OUTFILE	Writes the selected records to a file for user processing. See "Output Reports to File" on page 203 for more information.
PENDing	Valid for a schedule listing (LIST PLAN) only. Selects PENDING scheduled events.

PERManent	Selects events based on whether the events are defined as permanent. A permanent event occurs an unlimited number of times; a non-permanent event has a Times value which specifies a finite number of times.
	Examples:
	This command lists permanent events only:
	`LIST EVENT PERM=YES`
	This command lists non-permanent events only:
	`LIST EVENT PERM=NO`

PLan	Produces reports using the schedule queue records. You can enter one or more event numbers and use the RANGE, EXCLUDE, and EXRANGE parameters to select specific jobs. For current day schedule reports, schedule a LIST PLAN at the end of the day, immediately before the schedule load (make the schedule load a successor to the LIST PLAN job). For example, this command produces schedule listing: `LIST PLAN`

PRIority	Selects events that match the specified priority code (ranging from 1 through 15).
	Examples:
	This command lists events with a priority of 5:
	`LIST EVENTS PRIORITY 5`
	This command lists events with a priority of 7, 8, or 9:
	`LIST EVENTS PRI (7,8,9)`

Parameter	Description
RAnge	Specifies a range of events to select for reporting. You must specify a starting and ending event number within parentheses. More than one range can be entered. A starting number without an ending is assumed to go to the end of the database. For example, this command selects events from event number 40 through event 60, from event 100 through event 117, and from event 300 to the end: `LIST EVENTS RANGE (40,60,100,117,300)`
REAdy	Valid for a schedule listing (LIST PLAN) only. Selects events that are ready to run (time and WHEN conditions are satisfied), but have not been dispatched yet.
RECURring	Selects events that occur more than once within a schedule run (recurring events). For example, this command lists all recurring message events: `LIST EVENTS RECUR MSG`
REFreshed	Valid for a schedule listing (LIST PLAN) only. Selects events that were refreshed by the ZREFRESH command.

RESDETAIL	Selects all events in the database with logical resources defined.
	Examples:
	This command lists events with logical resources:
	`LIST EVENTS RESDETAIL`
	This command lists events with RESOURCE1 as a logical resource:
	`LIST EVENT RESDETAIL RESOURCE1`

REXX	Selects all REXX events. All event types are selected if no event type is specified. More than one event type can be entered on a command line.
	Examples:
	This command selects REXX events:
	`LIST EVENTS REXX`
	This command selects REXX and Zeke command events:
	`LIST EVENTS REXX ZCOM`

RUNNing	Valid for a schedule listing (LIST PLAN) only. Selects events that are currently running (i.e., have an Active event status). For example: `LIST PLAN APPL PAYROLL RUNNING`

Parameter	Description
SCOm	Selects all system command events. If entered alone, this selects all system command events. Can be combined on a command line with other event type parameters. (All event types are selected if no event type is specified.) For example, this command selects only system command events: `LIST EVENTS SCOM`
SECGroup	Selects job events with the specified security group (up to eight characters long).
SCHENV	Selects events with the specified scheduling environment requirement (up to 16 characters long). You can use wildcard and placeholder characters in your selection criteria (see "Generic Selection Criteria—Event/Plan Listings" on page 212). For example, this command selects events with that require the scheduling environment `PROD_DB_AVAIL`: `LIST EVENTS SCHENV PROD_DB_AVAIL`
STArt	Selects events scheduled after the specified time (in *hh*:*mm* format). The valid values range from `00:00` (default) through `47:59`. For example, this command selects events with a schedule time later than 04:00: `LIST EVENTS START 04:00`

STOp	Selects events scheduled earlier than the specified time (in *hh*:*mm* format). The valid values range from `00:01` to `47:59` (default).
	
	Examples:
	
	This command selects events that have a schedule time earlier than 16:00:
	
	`LIST EVENTS STOP 16:00`
	
	This command selects events with schedule times after 8:01 and before 16:00:
	
	`LIST EVENTS START 08:01 STOP 16:00`

SYStems	Selects events owned by the specified system (up to eight characters long). If this parameter is omitted, all systems and are selected.
	
	Examples:
	
	This command selects system A events:
	
	`LIST EVENTS SYS A`
	
	This command selects scheduled job events for systems A, MVSA, and MVSB:
	
	`LIST PLAN SYS (A, MVSA, MVSB) JOB`

Parameter	Description

TAPes	Selects job events that require at least one tape drive prior to job dispatch.
	Examples:
	This command lists job events that require at least one tape drive:
	`LIST EVENTS TAPE`
	This command lists job events for system B that require at least one tape drive:
	`LIST EVENTS TAPE SYS B`

TARGet	Selects events that execute on the specified remote system (up to eight characters long).
	Examples:
	This command lists scheduled events that execute on the remote system with a Netregid of REG4:
	`LIST PLAN TARGET REG4`
	This command lists events that execute on the remote system with a Netregid of Dallas45:
	`LIST EVENT TARG DALLAS45`

TASK	Selects task events that match the specified criteria. You can use wildcard and placeholder characters in your selection criteria (see "Generic Selection Criteria—Event/Plan Listings" on page 212).
	If entered alone, this selects all task events. You can limit the types of task events reported by specifying the type (e.g., sFTP).
	This parameter can be combined on a command line with other event type parameters. (All event types are selected if no event type is specified.)
	Examples:
	This command selects only task events:
	`LIST EVENTS TASK`
	This command selects only sFTP task events that have Zeke logins beginning with PR:
	`LIST EVENTS TASK SFTP LISTTASK LOGINN PR*`

TIMEOk	Valid for a schedule listing (LIST PLAN) only. Selects scheduled events that are time satisfied.

TITLe	Specifies the report title (up to 50 characters long) to be printed on the report heading. Enclose the title within delimiters. For example:
	`LIST PLAN TITLE 'SCHEDULE LISTING FOR SYSTEM X BY USERID'`
	`LIST EVENTS TITLE (EVENT MASTER LISTING BY SYSTEMID)`

Parameter	Description
TODay	Valid for a schedule listing (LIST PLAN) only. Selects only today's schedule records.

TRIGger	Selects recurring events that have at least one of the specified trigger code(s) to indicate when the event can satisfy WHEN conditions for other events. The valid values are A (all runs), F (first run), and L (last run). For example
	`LIST EVENT FIELDS=(TRIG EVENT TYPE ENAME JOB)`
	`SEQUENCE=(TRIG EVENT)`
	`TRIGGER=(F,L)`

USERid	Selects events with the specified user ID (up to eight characters long). For example, this command selects from the scheduled queue records (PLAN) all the jobs with user ID DATACTL1:
	`LIST PLAN USER DATACTL1`
	You can use wildcard and placeholder characters in your selection criteria (see "Generic Selection Criteria—Event/Plan Listings" on page 212).
	Examples:
	This command selects all events with a user ID beginning with DATA:
	`LIST EVENTS USER DATA*`
	This command selects events with a user ID beginning with D, ending with 1, and with any character in position 2:
	`LIST EVENTS USERID D?1`
	This command selects events with a user ID beginning with DA:
	`LIST EVENTS USER DA*`

VCOm	Selects all VM command events. For example: `LIST EVENTS VCOM` If entered alone, this selects all VM command events. Can be combined on a command line with other event type parameters. (All event types are selected if no event type is specified.)
WHendetail	Selects events with WHEN/SET conditions that contain the specified strings. You can enter up to 30 operands, using commas or spaces as separators. Multiple values are treated as OR conditions (if any of the strings are matched, the event is selected). If no values are specified, all events with a WHEN condition are selected. Mixed-case values are valid with this parameter. Strings containing spaces are not valid with this parameter.

Parameter	Description

	Examples:
	This command selects events that have a WHEN condition that contains $ABC in any position:
	`LIST EVENTS WHENDETAIL $ABC`
	This command selects events that have a NOTDURING JOB or NOTDURING PGM clause in the WHEN condition:
	`LIST EVENTS WHENDETAIL (NOTDURING)`
	This command selects events with any of the strings in the WHEN condition (that is, events with an EOJ, EOE, or variable in the WHEN condition):
	`LIST EVENTS WHENDETAIL (EOJ,VAR,EOE)`
	This command selects job events that have WHEN conditions with the jobnames PAY01P18 or PAY01P19, or a variable beginning with $VARXX:
	`LIST EVENTS JOB WHENDETAIL (PAY01P18,PAY01P19,$VARXX)`
	Only the number of characters entered for a value are compared. For example:
	`LIST EVENTS WHENDETAIL PAY1`
	This would match any WHEN condition with the string PAY1 in any position in the clause. For example:
	`EOJ PAY1BR14`
	`WEOJ WEAKPAY1`
	`VAR $ABC EQ DEVPAY1`
	For LIST PLAN, you can also select events based specifically on satisfied conditions. For example, this command selects events that have at least one satisfied variable WHEN condition:
	`LIST PLAN WHENDETAIL *VAR`
	Note:
	WHENDETAIL does not recognize wildcard characters.

WHENOK	Valid for a schedule listing (LIST PLAN) only. Selects scheduled events that are WHEN satisfied.

Parameter	Description
WORKcenter	Selects work center events. If entered alone, this selects all work center events. Can be combined on a command line with other event type parameters. (All event types are selected if no event type is specified.) For example, this command selects work center events only: `LIST EVENTS WORK`

ZCOm	Selects all Zeke command events.
	If entered alone, this selects all Zeke command events. Can be combined on a command line with other event type parameters. (All event types are selected if no event type is specified.)
	Examples:
	This command selects only Zeke command events:
	`LIST EVENTS ZCOM`
	This command selects job events and Zeke command events:
	`LIST EVENTS ZCOM JOB`

Sample Report —List Occurs, When and Variables for Job Events

Sample JCL

This sample JCL produces a sample LIST EVENTS report using the JOB, LISTOCCURS, LISTWHEN, and LISTVARS parameters:

```
//REPORT1 JOB
//STEP1 EXEC ZEKEUTL,P='SUBSYS=SSSI'
//SYSIN       DD  *
  LIST EVENTS JOB LISTOCCURS LISTWHEN LISTVARS
/*
```

Sample Report

This LIST EVENTS report was generated using the sample JCL:

```
EVENT  EVNT SCHED EARLY LATE  CALID    SYSTEM   APPLID   GRP USERID   EVENT NAME   JOB NAME    LAST      LAST
NUMBER TYPE TIME  TIME  START                             ID                                    DATE      TIME

   101 JOB  00:00   -     -    A       PRD                                          XXXXJOC   01/01/2017 12:26:30
            0 OCCURS (REQUEST)
   102 JOB  00:00   -     -    A       PRD                              PRD1        PRDSUB
            0 OCCURS (REQUEST)
          EVENT VARIABLE    VALUE
          $ACCTDD           'SYS2.ACCTING.PDOC'
          $RPTYPE           BASIC
   104 JOB  00:00   -     -    A       PRD                              PRD1        PRDJOBV   01/01/2017 18:31:30
            0 OCCURS (REQUEST)
   106 JOB  00:00   -     -    A       PRD                              VARIABLE    PRDJOBV
            0 OCCURS (REQUEST)
            0   WHEN (VAR $ABC EQ 10)
            1   WHEN (VAR $AAA2 EQ 20)
   110 JOB  00:00   -     -    A       PRD                              TVSET       TVSET     01/01/2017 18:31:30
            0 OCCURS (DAILY)
   112 JOB  00:00   -     -    A       PRD                              PRDJOB1     PRDJOB1
            0 OCCURS (DAILY)
   113 JOB  00:00   -     -    A       PRD      PAY1                    PRDJOB2     PRDJOB2
            0 OCCURS (DAILY)
   114 JOB  00:00   -     -    A       PRD      PAY1                    PRDJOB3     PRDJOB3
            0 OCCURS (DAILY)
   115 JOB  00:00   -     -    A       PRD      PAY1                    PRDJOB4     PRDJOB4   01/01/2017 18:31:30
            0 OCCURS (DAILY)
   116 JOB  00:00   -     -    A       PRD                                          PRDDATEJ  01/01/2017 18:31:30
            0 OCCURS (REQUEST)

       ******* NUMBER OF EVENTS REPORTED -          10
```

Sample Report —List Accounting Data for Job Events

Sample JCL

This JCL produces a sample LIST EVENTS report using the JOB and LISTACCTG parameters:

```
//REPORT1 JOB
//STEP1 EXEC ZEKEUTL,P='SUBSYS=SSSI'
//SYSIN       DD  *
  LIST EVENTS JOB LISTACCTG
/*
```

Sample Report

This LIST EVENTS report was generated using the sample JCL:

```
EVENT  EVNT SCHED EARLY LATE  CALID    SYSTEM   APPLID   GRP USERID   EVENT NAME   JOB NAME    LAST      LAST
NUMBER TYPE TIME  TIME  START                             ID                                    DATE      TIME
        5  JOB  00:00 -     -   DEFAULT  ASGX320A    COM DEVASG   IEFBR14    ASGBR14  01/01/2020 10:58:30
        DISPATCHED    23 TIMES   LAST UPDATE: 01/01/2020 15:33 BY DEVASG
        LAST DISPATCH:   START: 01/01/2020 AT 10:58  END: 01/01/2020 AT 10:58:00 JOB ID:
                      VERSION:     0         SCHED DATE: 01/10/2020            STATUS: SUCC
        DISPATCH DATE 01/01/2020 AT 10:58:00      TAPES:   0            VMEM:     0
        DUR: 00:00:00      CPUTIME: 00:00:00  COMP.CODE: C0000
        DISPATCH DATE 01/01/2020 AT 17:37:00      TAPES:   0            VMEM:     0
        DUR: 00:00:00      CPUTIME: 00:00:00  COMP.CODE: C0000
        DISPATCH DATE 01/01/2020 AT 11:31:00      TAPES:   0            VMEM:     0
        DUR: 00:00:00      CPUTIME: 00:00:00  COMP.CODE: C0000
        AVERAGE DURATION: 00:00:00
```

See “Status (EMR Accounting)” on page 498 for explanations of the status codes.

Sample Report -TASK EVENTS

This JCL produces a sample Task event report using the EVENT, TASKTYPE, ENAME, TARGET, and LOGINNAME parameters:

Sample JCL

```
//REPORT1  JOB
//STEP1    EXEC ZEKEUTL,P='SUBSYS=SSSI'
//SYSIN    DD   *
 LIST EVENT TASK LISTTASK
      FIELDS=(EVENT TASKTYPE ENAME TARGET LOGINNAME)
/*
```

Sample Report

This Task event report was generated using the sample JCL:

```
ZEKE 6.20 Z620A000 X320A000                                                    TUESDAY    DATE 01/01/2017  TIME 09:39:05
REPORT ZEKE02M-01  RUN 0260            ZEKE EVENT JOB RECORD LISTING                SEQ # 00001 PAGE      1
EVENT  TASK     EVENT NAME   SENDTO   LOGIN NAME
NUMBER TYPE
    42 Email    EMAIL_IT     UKZENA   ACME_EMAIL_UK
              SUBJECT: OVERNITE schedule failed
                   TO: schedulers@acme.co.uk
           IMPORTANCE: High
              MESSAGE: One of the jobs in the overnight schedule failed.
                       Please review the Schedule View entries for application OVERNITE.
                       Call Production Control at ext. 42 for assistance.
```

Sample Report—List Remote Task Events

Sample JCL

This JCL produces a sample LIST EVENTS report for remote task events:

```
//REPORT1 JOB
//STEP1 EXEC ZEKEUTL,P='SUBSYS=SSSI'
//SYSIN       DD  *
  LIST EVENT
    FIELDS=(EVENT TYPE TASKTYPE ENAME SYSTEM TARGET)
    LISTVARS
    LISTTASK
/*
```

Sample Report

This LIST EVENTS report was generated using the sample JCL:

```
EVENT  EVNT TASK     EVENT NAME   SYSTEM   SENDTO
NUMBER TYPE TYPE
     1 JOB  -                     DALLSYS1 *LOCAL
          EVENT VARIABLE    VALUE
          $VARIABLE1        today, tomorrow, never
          $VARIABLE2        you think so
          $VARIABLE3        a value here
     2 SCOM -                     DALLSYS1
    11 MSG  -                     DALLSYS1
    12 SCOM -                     DALLSYS1
    13 JOB  -                     DALLSYS1 *LOCAL
    14 TASK sFTP     TEST_SFTP_8  DALLSYS1
          HOSTNAME: dallas.ftp.server.com
          SFTPCMDS
            CHGDIR CHGLOCAL=Y LOCALDIR=(C:\Users\alice\Downloads) CHGREMOTE=Y REMOTEDIR=(/home/alice/Downloads)
            GET REMOTEFILE=(dailydata.xlsx) NEWLOCAL=Y LOCALFILE=(tuesday.xlsx) USEASCII=N
          CMDSSTOP
    15 PCOM -                     DALLSYS1
    16 REXX -                     DALLSYS1
    17 TASK sFTP     TEST_SFTP_9  DALLSYS1
          EVENT VARIABLE    VALUE
          $VARIABLE4        Red Hot
          $VARIABEL5        some text to be displayed somewhere
          $VARIABLE6        Diddly-squat
          HOSTNAME: ftworth.ftp.server.com
          SFTPCMDS
            CHGDIR CHGLOCAL=Y LOCALDIR=(C:\Users\alice\Downloads\Extended\) CHGREMOTE=Y REMOTEDIR=(/home/alice/Downloads/)
            GET REMOTEFILE=(monthlydata.xlsx) NEWLOCAL=Y LOCALFILE=(june.xlsx) USEASCII=N
          CMDSSTOP
       ******* NUMBER OF EVENTS REPORTED -          9
```

Note:

See “Status (EMR Accounting)” on page 498 for explanations of the status codes.

Sample Report —List Documentation for Scheduled Events

Sample JCL

This JCL produces a sample LIST PLAN report for system "PRD" using the SYS, LISTNOTES, LISTPAD, and LISTTEXT parameters:

```
//REPORT3 JOB
//STEP1 EXEC ZEKEUTL,P='SUBSYS=SSSI'
//SYSIN       DD  *
  LIST PLAN SYS PRD LISTNOTES LISTPAD LISTTEXT
/*
```

Sample Report

This LIST PLAN report was generated using the sample JCL:

```
EVENT  VERSN EVNT SCHED EARLY LATE  CALID    SYSTEM   APPLID   GRP USERID   EVENT NAME   JOB NAME    LAST      LAST
NUMBER       TYPE TIME  TIME  START                             ID                                    DATE      TIME

     3     0 WORK 00:00   -     -   A        PRD                            WORKCTR              01/19/2017 08:53:30

 COMMENT LINES : THIS IS A WORK CENTER EVENT

    10     0 JOB  00:00   -     -    A       PRD                            TVSET        TVSET    01/09/2017 09:04:30
    12     0 JOB  00:00   -     -    A       PRD                            PRDJOB1      PRDJOB1
    13     0 JOB  00:00   -     -    A       PRD                            PRDJOB2      PRDJOB2
    14     0 JOB  00:00   -     -    A       PRD                            PRDJOB3      PRDJOB3
    15     0 JOB  00:00   -     -    A       PRD      PAY1                  PRDJOB4      PRDJOB4  01/09/2017 09:04:30
    16     0 JOB  00:00   -     -    A       PRD                                         PRDDATEJ 01/28/2017 18:18:30
    16     0 JOB  00:00   -     -    A       PRD                                         PRDDATEJ 01/28/2017 18:18:30
    16     0 JOB  00:00   -     -    A       PRD                                         PRDDATEJ 01/28/2017 18:18:30
     7     0 WORK 00:00   -     -    A       PRD                            MULTIVAR             01/15/2017 07:48:30

 COMMENT LINES : THIS HAS MULTIPLE VARIABLES

   SCRATCH PAD - THIS IS FOR QUICK NOTES.

      TEXT DOC - THIS IS THE TEXT INFORMATION.                                          00010000
                                                                                        00011000
                 IT MAY BE USED FOR LONG DESCRIPTIONS THAT COULD SPAN SEVERAL LINES.    00020000

      NOTE DOC - THIS IS A NOTE
     8     0 WORK 00:00   -     -    A       PRD      AP1                  EVTAP1

 COMMENT LINES : THIS EVENT IS FOR AP1
     9     0 WORK 00:00   -     -    A       PRD      AP2                  EVTAP2

 COMMENT LINES : THIS EVENT IS FOR AP2
    11     0 WORK 00:00   -     -    A       PRD

 COMMENT LINES : THIS WORK CENTER USES '>' AND '<'

        ******* NUMBER OF EVENTS REPORTED -          14
```

Sample Report —List Uncompleted Scheduled Events

Sample JCL

This JCL produces a LIST PLAN report using the NOTDONE, TITLE, SPACE2, FIELDS, SEQUENCE, LISTW, and LISTDOC parameters, sorted by the schedule time:

```
//REPORT5 JOB
//STEP1 EXEC ZEKEUTL,P='SUBSYS=SSSI'
//SYSIN       DD  *
  LIST PLAN NOTDONE TITLE 'DAILY SCHEDULE OF EVENTS'
  SPACE2 FIELDS
  (SCHED,EVENT,ENAME,JOBNAME,DESC,OPEROK,
  AVGDUR) SEQUENCE (SCHED) LISTW LISTDOC
/*
```

Sample Report

This LIST PLAN report was generated using the sample JCL:

```
SCHED EVENT  EVENT NAME   JOB NAME DESCRIPTION OF EVENT                                        OPR  AVGDUR
TIME  NUMBER                                                                                  OK  HH:MM:SS

00:00      7 MULTIVAR              MULTI VARIABLE WORK CENTER                                 NO
              0   WHEN (?XVAR A1 EQ 10 AND ?XVAR A2 EQ 20 AND XVAR A3 EQ 30 AND XVAR A4 EQ 40)

   COMMENT LINES : THIS HAS MULTIPLE VARIABLES

   SCRATCH PAD - THIS IS FOR QUICK NOTES.

      TEXT DOC - THIS IS THE TEXT INFORMATION.
                 IT MAY BE USED FOR LONG DESCRIPTIONS THAT COULD SPAN SEVERAL LINES.

      NOTE DOC - THIS IS A NOTE

00:00      8 EVTAP1                THIS IS AP1 EVENT                                          NO
              0   WHEN (?XVAR APPL1 EQ EVT1)

 COMMENT LINES : THIS EVENT IS FOR AP1

00:00      9 EVTAP2                THIS IS AP2 EVENT                                          NO
              0   WHEN (?XVAR APPL1 EQ EVT2)

 COMMENT LINES : THIS EVENT IS FOR AP2

00:00     10 TVSET        TVSET    THIS JOB IS BASED ON A VARIABLE VALUE                      NO  00:00:01

00:00     11                       WORK CENTER                                                NO
              0   WHEN (?XVAR A1 EQ 10)

 COMMENT LINES : THIS WORK CENTER USES '>' AND '<'

00:00     17              MAAZJOB2 MATTS 2                                                    NO  00:00:00

05:00     15 PRDJOB4      PRDJOB4  THIS JOB RETURNS RESULT 4                                 NO  00:00:00

06:00     14 PRDJOB3      PRDJOB3  THIS JOB RETURNS RESULT 3                                 NO  00:00:00

07:00     13 PRDJOB2      PRDJOB2  THIS JOB RETURNS RESULT 2                                 NO  00:00:00

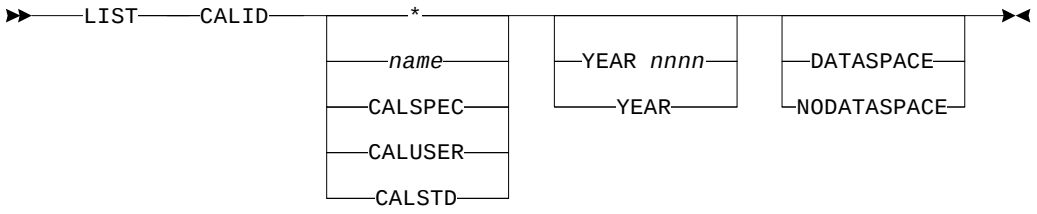
08:00     12 PRDJOB1      PRDJOB1  THIS JOB RETURNS RESULT 1                                 NO  00:00:00

       ******* NUMBER OF EVENTS REPORTED - 10
```

LIST CALID—Zeke Calendar Information

You use this command to list information about the system calendars. Select one or more calendars by specifying the calendar IDs.

Syntax



Parameters

These are the valid selection and report generation parameters:

Parameter	Description
*	Reports on all existing calendars.

name	Reports on the specified calendar for the current year.
	Examples:
	This command reports on the Q1 calendar:
	`LIST CALID Q1`
	This command reports on ACCT1, ACCT2, and ACCT3 calendars for the current year:
	`LIST CALID (ACCT1,ACCT2,ACCT3)`
	This command reports on calendars A, NORMAL, and F:
	`LIST CALID (A,NORMAL,F)`

CALSPEC	Reports on all special calendars.
CALUSER	Reports on all user calendars.
CALSTD	Reports on all standard calendars.

Parameter	Description
DATASPACE	Indicates to use a data space when generating the report. **Note:** If you always want to use a data space for generating reports, set the DSPBatch generation option to Y (see page 541). `LIST CALID Q1 DATASPACE`
NODATASPACE	If the DSPBatch generation option is set to Y (see page 541), the ZEKE utility program uses a data space when generating reports. To override this for a particular backup so that a data space is not used, use the NODATASPACE parameter.
YEAR	Reports on a specific calendar year. If omitted, the current year calendar is selected. For example, this command reports on the calendar ACCT2 for 2020: `LIST CALID ACCT2 YEAR 2020`

Sample Report—LIST CALID

Sample JCL

This JCL produces a LIST CALID report:

```
//REPORT6 JOB
//STEP1 EXEC ZEKEUTL,P='SUBSYS=SSSI'
//SYSIN       DD  *
  LIST CALID A
/*
//
```

Sample Report

This LIST CALID report was generated using the sample JCL:

```
ZEKE 6.2 Z620A000 X320A000 - ZEKE UTILITY PROGRAM MAINTENANCE LISTING SSSI  RUN 0039  DATE 01/23/2020

  LIST CALID A

Z02C8I REPORT SEQUENCE NUMBER 001 - RECORDS SELECTED=0000001
Z0214I ENTERING SORT PHASE
Z0215I ENTERING PRINT PHASE


                ZEKE STANDARD CALENDAR ID: A         YEAR: ****

** WORK DAYS **          ****************** HOLIDAYS *****************

   MONDAY: YES           01/01/****  07/04/****  12/25/****  12/31/****
  TUESDAY: YES
WEDNESDAY: YES
 THURSDAY: YES
   FRIDAY: YES
 SATURDAY: NO
   SUNDAY: NO

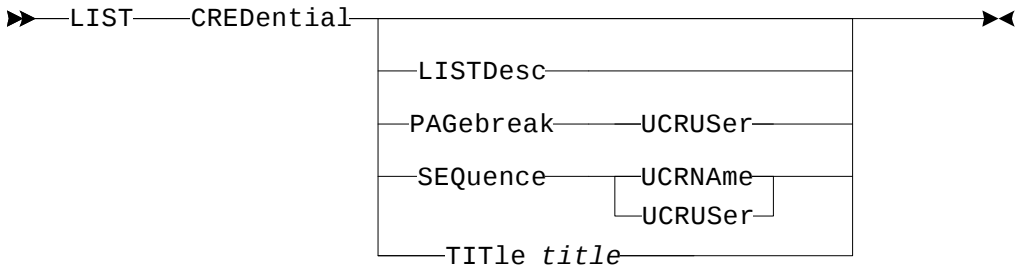
    FISCAL START MONTH: 01
   CALENDAR START DATE:              CALENDER END DATE:
  CALENDAR EXPIRE DATE:             DATE LAST ACCESSED: 01/09/2020


  ******* NUMBER OF CALENDARS REPORTED -       1
```

LIST CREDENTIAL—Zeke Credentials

You use this command to list Zeke credentials.

Syntax



Parameters

These are the valid report generation parameters:

Note:

The FIELDS and SPACE*n* parameters are not valid with the LIST CREDENTIAL command.

Parameter	Description
`LISTDesc`	Prints the credential description (if the credential definition contains one) on the report.
`PAGebreak`	Indicates the field that controls page breaks on the report. Use this keyword with the PAGEBREAK parameter: `UCRUSer` Inserts a page break at each new user ID.
`SEQuence`	Indicates the sort sequence for printing credentials on the report. These are the valid keywords: `UCRNAme` Sorts by credential name. `UCRUSer` Default. Sorts by user ID.
`TITle` *`title`*	Prints the specified *`title`* as the report heading. Use delimiters if the title contains more than one word.

Sample JCL

This sample JCL produces a LIST CREDENTIAL report:

```
//REPORT7 JOB
//STEP1 EXEC ZEKEUTL,P='SUBSYS=SSSI'
//SYSIN       DD  *
  LIST CRED LISTD PAG UCRUS TITLE 'CURRENT USER CREDENTIALS'
/*
//
```

Sample Report

This sample illustrates a the LIST CREDENTIAL report:

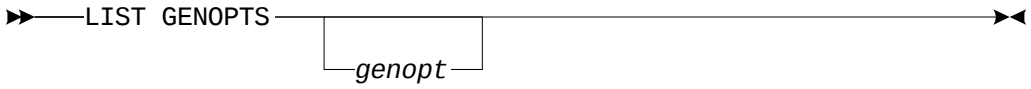
```
ZEKE 6.20 Z620A000 X320A000                ACME SERVICES CORP              WEDNESDAY   DATE MM/DD/YYYY  TIME 13:21:13
REPORT ZEKE02M-10  RUN 0057                  ZEKE CREDENTIAL DEFINITIONS LISTING           SEQ # 001   PAGE      1
CREDENTIAL NAME                 USER ID
ALICE_SERVERS                   alice.smith@acme.corp
  DESCRIPTION: Used for logging in to the general servers.
BOB_MAINFRAME                   BOB
```

LIST GENOPTS—Zeke Generation Options

You use this command to list details for a Zeke GENOPT (e.g., brief description, fields and their current values, and last modification information).

You can generate a report based on the `*ACTIVE` GENOPT (i.e., generation option values currently in memory) to help with troubleshooting.

Syntax



Parameter

You can use this parameter for selecting a GENOPT for the report:

Parameter	Description
genopt	Specifies the name of the local GENOPT to be reported, or the `*GLOBAL` or `*ACTIVE` GENOPT. If omitted, the default local GENOPT (`********`) (loaded at Zeke startup) is reported. **Note:** Because a GENOPT name (e.g., special GENOPTs, `*ACTIVE` and `*GLOBAL`) can contain an asterisk (`*`), you cannot use wildcard characters.

Sample Report

Sample JCL

This JCL produces a LIST GENOPTS report:

```
//REPORT7 JOB
//STEP1 EXEC ZEKEUTL,P='SUBSYS=SSSI'
//SYSIN       DD  *
  LIST GENOPTS
/*
//
```

Sample Report

This sample illustrates a segment of the LIST GENOPTS report for the default local GENOPT:

```
REPORT ZEKE02M-08  RUN 0004                  ZEKE GENERATION OPTIONS ...

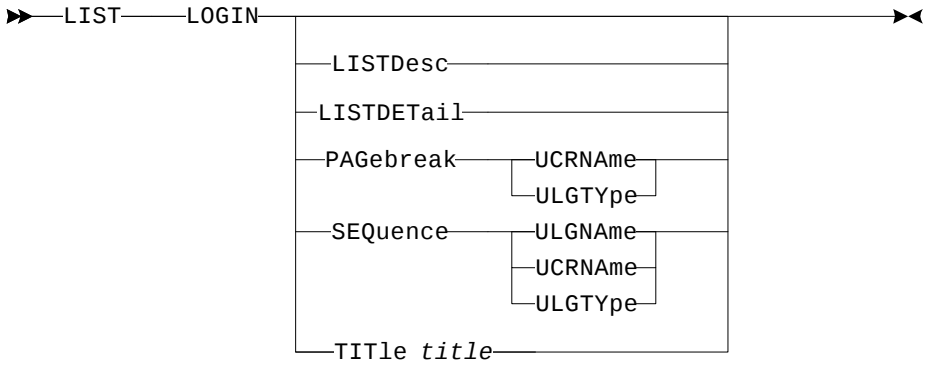
ZEKE SYSTEM: ********  DESCRIPTION: Default local system options
LAST UPDATED: ASG      08/13/2017 16:35:04

OPTION    VALUE             DESCRIPTION
Aur       Y                 Yes to enable automatic responses
AurIntv   1                 Number of seconds to check auto responses
AurMsg    Y                 Yes to inform operator auto. response issues
BimAppl                     Bimedit application name
...
```

LIST LOGIN—Zeke Logins

You use this command to list Zeke logins.

Syntax



Parameters

These are the valid report generation parameters:

Note:

The FIELDS and SPACE*n* parameters are not valid with the LIST LOGIN command.

Parameter	Description
`LISTDesc`	Prints the login description (if the login definition contains one) on the report.
`LISTDETail`	Prints detailed information for each login on the report.
`PAGebreak`	Indicates the field that controls page breaks on the report. These are the valid keywords: `UCRNAme` Inserts a page break at each new credential name. `ULGTYpe` Inserts a page break at each new login type.
`SEQuence`	Indicates the sort sequence for printing logins on the report. These are the valid keywords: `UCRNAme` Sorts by credential name. `ULGNAme` Default. Sorts by login name.

Parameter	Description
	`ULGTYpe` Login type.
`TITle title`	Prints the specified *`title`* as the report heading. Use delimiters if the title contains more than one word.

Sample JCL

This sample JCL produces a LIST LOGIN report:

```
//REPORT7 JOB
//STEP1 EXEC ZEKEUTL,P='SUBSYS=SSSI'
//SYSIN       DD  *
  LIST LOGIN LISTD SEQ UCRNA TITLE 'CURRENT LOGINS'
/*
//
```

Sample Report

This sample illustrates a LIST LOGIN report:

```
ZEKE 6.20 Z620A000 X320A000                                                    WEDNESDAY   DATE 02/01/2017 TIME 14:22:56
REPORT ZEKE02M-10 RUN 0101              ZEKE LOGIN DEFINITIONS LISTING                      SEQ # 00001 PAGE      1
LOGIN NAME                      LOGIN TYPE       CREDENTIAL NAME
ACME_FTP_LOGIN                  sFTP             ACME_FTP_CREDENTIAL
CA_EMAIL_LOGIN                  Email            CA_EMAIL_CREDENTIAL
NORWAY_EMAIL                    Email            EMAIL_CREDENTIAL
```

Sample Report- LIST LOGIN LISTDETAIL

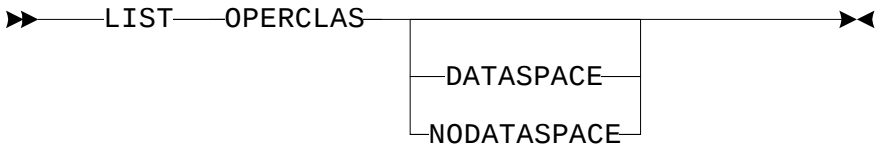
This sample illustrates a LIST LOGIN LISTDETAIL report:

```
ZEKE 6.20 Z620A000 X320A000                                                    WEDNESDAY   DATE 02/01/2017 TIME 14:22:56
REPORT ZEKE02M-10 RUN 0101              ZEKE LOGIN DEFINITIONS LISTING                      SEQ # 00001 PAGE      1
LOGIN NAME                      LOGIN TYPE       CREDENTIAL NAME
ACME_FTP_LOGIN                  sFTP             ACME_FTP_CREDENTIAL
  DESCRIPTION: Login for Acme general FTP servers
     HOSTNAME: ftp.acme.com
     HOSTTYPE: Unix
    PROTOTYPE: SFTP_USING_SSH2
         PORT: 22
     COMPRESS: Y
      PASSIVE: Y
      USECERT: Y
   ALLOWTOKEN: Y
CA_EMAIL_LOGIN                  Email            CA_EMAIL_CREDENTIAL
  DESCRIPTION: Login for Canada region email servers
     HOSTNAME: mail.acme.com.ca
         PORT: 25
  FROMADDRESS: Zeke SYSA <sysa.mainframe@acme.com.ca>
     REQLOGIN: Y
NORWAY_EMAIL                    Email            EMAIL_CREDENTIAL
  DESCRIPTION: Login data for Norway email servers
     HOSTNAME: mail.acme.com.no
         PORT: 25
  FROMADDRESS: Zeke SYSA <sysa_mainframe@acme.com>
     REQLOGIN: Y
```

LIST OPERCLAS—Operator ID Security Classes

You use this command to list information about all operator ID security classes defined in the database.

Syntax



Parameters

These are the valid report generation parameters:

Parameter	Description
DATASPACE	Use a data space when generating the report. **Note:** If you always want to use a data space for generating reports, set the DSPBatch generation option to Y (see page 541). For example: `LIST OPERCLAS DATASPACE`
NODATASPACE	If the DSPBatch generation option is set to Y (see page 541), the ZEKE utility program uses a data space when generating reports. To override this for a particular backup so that a data space is not used, use the NODATASPACE parameter.

Sample Report—LIST OPERCLAS

Sample JCL

This JCL produces a LIST OPERCLAS report:

```
//REPORT11 JOB
//STEP1 EXEC ZEKEUTL,P='SUBSYS=SSSI'
//SYSIN       DD  *
  LIST OPERCLAS
/*
//
```

Sample Report

This LIST OPERCLAS report was generated using the sample JCL:

```
ZEKE 6.2 Z620ADV6 X320A000 - ZEKE UTILITY PROGRAM MAINTENANCE LISTING   SSSI  RUN 0117

LIST OPERCLAS

Z02C8I REPORT SEQUENCE NUMBER 002 - RECORDS SELECTED=0000002
Z0214I ENTERING SORT PHASE
Z0215I ENTERING PRINT PHASE

ZEKE 6.2 Z620ADV6 X320A000
REPORT ZEKE02M-03  RUN 0117              ZEKE SECURITY CLASS LISTING

CLASS: A  Evt- W  Zcm- W  Cal- W  Opt- W  Wrk- W  Sec- W  Doc- W  Var- W  Rst- W
Schedule Control (Operator) Commands Allowed (Y=yes, N=no)

Zadd       -  Y         Zid        -  Y         Zreload    -  Y         Zres Disp  -  Y
Zalter     -  Y         Zkill      -  N         Zset       -  Y         Zres Alt   -  Y
Zdelete    -  N         Zmap       -  Y         Zstatus    -  Y         Zenable    -  N
Zdisable   -  Y         Zhold      -  Y         Zrelease   -  Y         Zrefresh   -  Y
Zdisplay   -  Y         Zok        -  Y         Zscan      -  Y         Zres Rel   -  Y


CLASS: B  Evt- W  Zcm- W  Cal- W  Opt- W  Wrk- W  Sec- W  Doc- W  Var- W  Rst- W
Schedule Control (Operator) Commands Allowed (Y=yes, N=no)

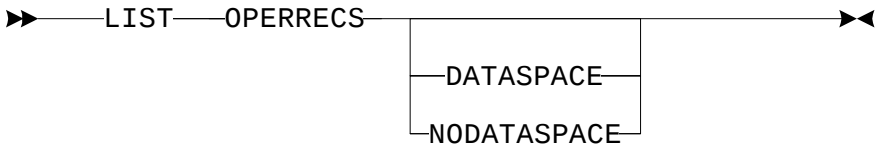
Zadd       -  N         Zid        -  Y         Zreload    -  Y         Zres Disp  -  Y
Zalter     -  Y         Zkill      -  N         Zset       -  Y         Zres Alt   -  Y
Zdelete    -  N         Zmap       -  Y         Zstatus    -  Y         Zenable    -  Y
Zdisable   -  N         Zhold      -  Y         Zrelease   -  N         Zrefresh   -  Y
Zdisplay   -  Y         Zok        -  Y         Zscan      -  Y         Zres Rel   -  Y

   **** Number of Security classes reported -      2
```

LIST OPERRECS—Zeke Operator ID Information

You use this command to list information about all operator IDs defined in the database.

Syntax



Parameters

These are the valid report generation parameters:

Parameter	Description
DATASPACE	Use a data space when generating the report. **Note:** If you always want to use a data space for generating reports, set the DSPBatch generation option to Y (see page 541). For example: `LIST OPERRECS DATASPACE`
NODATASPACE	If the DSPBatch generation option is set to Y (see page 541), the ZEKE utility program uses a data space when generating reports. To override this for a particular backup so that a data space is not used, use the NODATASPACE parameter.

Sample Report—LIST OPERRECS

Sample JCL

This JCL produces a LIST OPERRECS report:

```
//REPORT10 JOB
//STEP1 EXEC ZEKEUTL,P='SUBSYS=SSSI'
//SYSIN       DD  *
  LIST OPERRECS
/*
//
```

Sample Report

This LIST OPERRECS report was generated using the sample JCL:

```
ZEKE 6.2 Z620ADV6 X320A000 - ZEKE UTILITY PROGRAM MAINTENANCE LISTING   SSSI  RUN 0117

LIST OPERRECS

Z02C8I REPORT SEQUENCE NUMBER 001 - RECORDS SELECTED=0000003
Z0214I ENTERING SORT PHASE
Z0215I ENTERING PRINT PHASE

ZEKE 6.2 Z620ADV6 X320A000
REPORT ZEKE02M-03  RUN 0117              ZEKE OPERATOR ID LISTING

Operator Id: CARLTST    Class Id: A  Date Added: 01/01/2020  Last Updte: 01/01/2020
Userid:             Zcom     Event      Work    Documentation    Variable
********              W        W          W          W              W
CCCCCCC               W        W          W          W              W
ALLGOV                W        W          W          W              W
SEINI*                W        W          W          W              W
BARBARA               W        W          W          W              W
STEVEW*               W        W          W          W              W
RICHARDS              W        W          W          W              W

Operator Id: OPERATOR   Class Id: A  Date Added: 01/01/2020  Last Updte:
Userid:             Zcom     Event      Work    Documentation    Variable
********              W        W          W          W              W
ALLPAY*               W        W          W          W              W

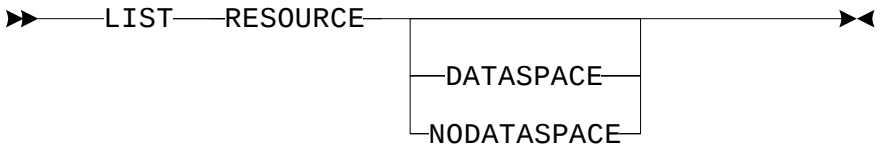
Operator Id: TEST1      Class Id: A  Date Added: 01/01/2020  Last Updte: 01/01/2020
Userid:             Zcom     Event      Work    Documentation    Variable
********              W        W          W          W              W
USER2                 N        N          N          W              W
USER3                 W        W          W          N              N
USER4                 W        W          W          N              N

  ******* NUMBER OF OPERATOR IDS REPORTED -       3
```

LIST RESOURCE—System Resources

You use this command to list the resource name and description.

Syntax



Parameters

These are the valid report generation parameters:

Parameter	Description
DATASPACE	Use a data space when generating the report. **Note:** If you always want to use a dataspace for generating reports, set the DSPBatch generation option to Y (see page 541). `LIST RESOURCE DATASPACE`
NODATASPACE	If the DSPBatch generation option is set to Y (see page 541), the ZEKE utility program uses a data space when generating reports. To override this for a particular backup so that a data space is not used, use the NODATASPACE parameter.

Sample Report—LIST RESOURCE

Sample JCL

This JCL produces a LIST RESOURCE report:

```
//REPORT12 JOB
//STEP1 EXEC ZEKEUTL,P='SUBSYS=SSSI'
//SYSIN       DD  *
  LIST RESOURCE
/*
//
```

Sample Report

This LIST RESOURCE report was generated using the sample JCL:

```
ZEKE 6.2 Z620A000 X320A000     THIS IS THE DATABASE FOR GWS.....  TUESDAY  DATE 01/01/2020
TIME 12:48:07
REPORT ZEKE02M-03  RUN 0148                LIST RESOURCES ON TSO45          SEQ # 005  PAGE    1

Resource name                                    System          Maximum   Active?
                                                                 Shared
TAPEDRIVE                                        OPDESK          0001      YES
EDRESO                                           (GLOBAL)        0100      YES
EDR1                                             TSO45           0001      YES
EDR2                                             MEDA            0001      YES
EDR2                                             TSO45           0001      YES
EDR3                                             (GLOBAL)        0001      YES
TAPE                                             (GLOBAL)        9901      YES
NEWRESO                                          (GLOBAL)        0001      YES
ALPHA                                            (GLOBAL)        0001      YES
TSO45                                            (GLOBAL)        0001      YES
REG7                                             TSO45           0010      YES
REG8                                             TSO45           0010      YES

******* NUMBER OF RESOURCE NAMES REPORTED -      12
```

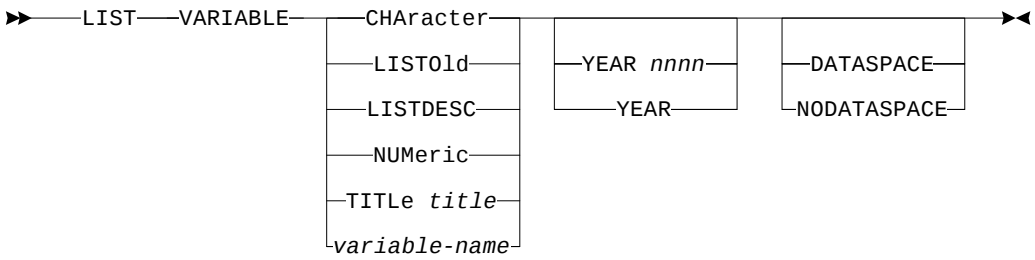
LIST VARIABLE—Zeke Variable Information

You use this command to list Zeke variables. When system-dependent variables are specified in a selection list, the names are resolved for the system on which the report is being produced.

Note:

The LIST VARIABLE report does not list OASIS variables.

Syntax



Parameters

These are the valid report generation parameters:

Parameter	Description
CHAracter	Selects only variables with character formats.
DATASPACE	Use a data space when generating the report. **Note:** If you always want to use a dataspace for generating reports, set the DSPBatch generation option to Y (see page 541). For example: `LIST VARIABLE NUM DATASPACE`
LISTOLd	Lists the previous and current values of the selected variables.
LISTDESC	Lists the descriptions of the selected variables.
NODATASPACE	If the DSPBatch generation option is set to Y (see page 541), the ZEKE utility program uses a data space when generating reports. To override this for a particular backup so that a data space is not used, use the NODATASPACE parameter.

Parameter	Description
`NUMeric`	Selects only variables with numeric formats.
`TITLe`	Specifies the report title (up to 50 characters long) to be printed on the report heading. For example: `LIST VARIABLE CHAR TITLE (CHARACTER FORMAT ZEKE VARIABLES)`

`variable-name`	Selects the indicated variables. If no variables are listed, selects all system variables defined in the catalog. Up to 30 variables can be specified for each report.
	Examples:
	This command lists all Zeke variables:
	`LIST VARIABLE`
	This command lists only variable $XYZ:
	`LIST VAR $XYZ`
	This command lists these four variables:
	`LIST VAR ($A,$B,$C,$D)`
	Note:
	To perform wildcard selection for a variable name, precede the variable with an asterisk (*) and enter only the characters to be compared. An asterisk in any other position is considered to be a placeholder character.
	This command lists variables beginning with $PAY:
	`LIST VAR (*$PAY)`
	This command lists variables with $A in the first and second positions and 01 in the sixth and seventh positions:
	`LIST VAR ($A***01)`

Sample Report 1—LIST VARIABLES

Sample JCL

This JCL produces a sample LIST VARIABLES report using the TITLE parameter:

```
//REPORT8 JOB
//STEP1 EXEC ZEKEUTL,P='SUBSYS=sssi'
//SYSIN       DD  *
  LIST VARIABLES TITLE 'CURRENT VARIABLES'
/*
//
```

Sample Report

This LIST VARIABLES report was generated using the sample JCL:

```
ZEKE 6.2 Z620A000 X320A000 - ZEKE UTILITY PROGRAM MAINTENANCE LISTING SSSI  RUN 0038   DATE  01/01/2020

  LIST VARIABLES TITLE 'CURRENT VARIABLES'

Z02C8I REPORT SEQUENCE NUMBER 001 - RECORDS SELECTED=0000006
Z0214I ENTERING SORT PHASE
Z0215I ENTERING PRINT PHASE


ZEKE 6.2 Z620A000 X320A000                                          THURSDAY   DATE 01/01/2020  TIME 15:19:23
REPORT ZEKE02M-03  RUN 0038                       CURRENT VARIABLES                        SEQ # 001   PAGE   1

ZEKE DATA-NAME   DATE SET   TIME SET  JOB/PGM NAME  PR/SYSTEM     DATA-NAME VALUE

$AAA1            01/01/2020 10:04:23  OPERATOR USR TSO-PRD        NUMERIC VALUE = 10
$AAA2            01/01/2020 10:04:24  OPERATOR USR TSO-PRD        NUMERIC VALUE = 20
$AAA3            01/01/2020 10:04:24  OPERATOR USR TSO-PRD        NUMERIC VALUE = 30
$AAA4            01/01/2020 10:04:24  OPERATOR USR TSO-PRD        NUMERIC VALUE = 40
$ABC             01/01/2020 14:50:53  PDOPER   USR TSO-PRD        GHI
$DEF             01/01/2020 08:23:25  PRD1     USR TSO-PRD        XYZ


  ******* NUMBER OF DATA-NAMES REPORTED -      6
```

Sample Report 2—LIST VARIABLES

Sample JCL

This JCL produces a LIST VARIABLES report using the LISTOLD parameter:

```
//REPORT9 JOB
//STEP1 EXEC ZEKEUTL,P='SUBSYS=SSSI'
//SYSIN       DD  *
  LIST VAR LISTOLD
/*
//
```

Sample Report

This LIST VARIABLES report was generated using the sample JCL:

```
ZEKE 6.2 Z620A000 X320A000 - ZEKE UTILITY PROGRAM MAINTENANCE LISTING SSSI  RUN 0037   DATE  01/01/2020

  LIST VAR LISTOLD

Z02C8I REPORT SEQUENCE NUMBER 001 - RECORDS SELECTED=0000006
Z0214I ENTERING SORT PHASE
Z0215I ENTERING PRINT PHASE


ZEKE 6.2 Z620A000 X320A000                                          THURSDAY   DATE 01/01/2020  TIME 15:15:13
REPORT ZEKE02M-03  RUN 0037                 ZEKE DATA-NAME LISTING                        SEQ # 001   PAGE   1

ZEKE DATA-NAME   DATE SET   TIME SET  JOB/PGM NAME  PR/SYSTEM     DATA-NAME VALUE

$AAA1            01/25/2020 10:04:23  OPERATOR USR TSO-PRD        NUMERIC VALUE = 10
    PRIOR VALUE               :  :            JOB    -          ---- NO PREVIOUS VALUE EXISTS -----
$AAA2            02/25/2020 10:04:24  OPERATOR USR TSO-PRD        NUMERIC VALUE = 20
    PRIOR VALUE  02/13/2020 09:12:10  OPERATOR USR TSO-PRD        NUMERIC VALUE = 20
$AAA3            02/25/2020 10:04:24  OPERATOR USR TSO-PRD        NUMERIC VALUE = 30
    PRIOR VALUE  02/13/2020 09:12:11  OPERATOR USR TSO-PRD        NUMERIC VALUE = 30
$AAA4            02/25/2020 10:04:24  OPERATOR USR TSO-PRD        NUMERIC VALUE = 40
    PRIOR VALUE  02/13/2020 09:12:12  OPERATOR USR TSO-PRD        NUMERIC VALUE = 40
$ABC             02/23/2020 14:50:53  PDOPER   USR TSO-PRD        GHI
    PRIOR VALUE  01/17/2020 10:23:28  PRD1     USR TSO-PRD        DEF
$DEF             01/16/2020 08:23:25  PRD1     USR TSO-PRD        XYZ
    PRIOR VALUE  01/16/2020 08:23:20  PRD1     USR TSO-PRD        ---- NO PREVIOUS VALUE EXISTS -----


  ******* NUMBER OF DATA-NAMES REPORTED -       6
```

5 Operator Commands

This chapter describes the Zeke operator commands. It discusses these topics:

Topic	Page
Issuing Zeke Operator Commands	257
Command Prefix	257
Multiple Commands	258
Text Conventions	258
Command Abbreviations	258
Command Syntax	259
Command Acceptance	262
Command Authentication	262
Global Parameters	263
ZADD Command	276
ZALTER Command	288
Altering JCL Source	288
Altering Other Event Fields	289
Altering Message Generation	299
Altering Initiator Availability	300
ZDELETE Command	302
ZDISABLE Command	310
Disabling Scheduled Events	310
Disabling Automatic Replies	311
Disabling Electronic Vaulting	312
ZDISPLAY Command	313
ZDISPLAY Formats	313
Displaying Event Information	313
Displaying Preceding and Succeeding Events (PathFinder)	334
Displaying Variables	337
Displaying Initiator Information	338
Displaying Automatic Reply Information	342
Displaying System Pool Information	343
Displaying Zeke Agents	343
Displaying Remote Agents	345
Displaying Remote Events, Variables and Triggers	347

Topic	Page
Displaying Communication Record Information	350
Displaying Database Information	350
Displaying Generation Options	351
Displaying Tracing Messages	353
ZENABLE Command	362
ZHOLD Command	363
ZID Command	367
ZINFO Command	368
ZKILL Command	369
ZMAP Command	372
ZOK Command	374
ZPLEX Command	375
ZREFRESH Command	385
ZRELEASE Command	386
ZRELOAD Command	390
ZRESOURCE Command	393
ZSCAN Command	397
ZSET Command	399
Setting Variable Values	399
Setting the Default System ID	401
ZSTATUS Command	402
Zeke Address Space Commands	403
#APPEND	404
#CLEAR	405
#DISPLAY	406
#RESET	407
#SET 408	
#START	409
Zeke Server Commands	410
$CLOSE	410
$DISPLAY	412
$KILL	418
$OPEN	421
$SET	422

Issuing Zeke Operator Commands

You can issue the Zeke operator commands using any of these methods:

- From the system console or SDSF. Commands are executed through the Zeke started task command processor.
- Through the /ZCOM function (see the *ASG-Zeke Scheduling for z/OS User's Guide* for details).
- From any command line in the Zeke ISPF online facility (with the exception of the ZKILL command). Commands are executed in the TSO user's address space.
- From a command shell. You can issue the ZCOM command from a Zeke ISPF command line to open a command shell where you can issue commands up to 200 characters long. The command shell also enables you to retrieve the last ten commands that were issued (from the command shell).
- Using a Zeke host command (i.e., a REXX ADDRESS ZEKE ZCOM command). Commands are executed in the address space where REXX is running (i.e., batch, ISPF, or OASIS/ECF).
- Using a ZEKESET command (i.e., SET SCOM or SET ZCOM). Commands are executed depending on how they are specified. In a SET SCOM function, the Zeke command is issued to the operator console and executed in the Zeke started task command processor. In a SET ZCOM function, the Zeke command is executed through the batch job that runs the ZEKESET program.
- Through an SCOM or ZCOM event. Commands are executed depending on how they are defined in the command event. In an SCOM event, the Zeke command is issued to the operator console and executed in the Zeke started task command processor. In a ZCOM event, the Zeke command is executed through the Zeke dispatcher.
- Using the ZEKECMD API. Commands are executed in the same environment that calls ZEKECMD.
- From an OpsCentral console (as a raw Zeke command). Commands are executed through the Zeke server address space.

Command Prefix

If you are running more than one Zeke, you must add an additional prefix to the Zeke command to distinguish between systems. Use the CMDPREFX parameter in your OASIS options to add a command prefix. See the *ASG-OASIS for z/OS Reference Guide* for additional information.

Multiple Commands

You can enter multiple Zeke commands on the command line simply by entering a command, then following it with a second command, and so on.

When entering multiple Zeke operator commands on a single line, the command prefix is needed only on the first command. Including the prefix on the additional command(s) could generate an operand length error, resulting in none of the commands being executed.

For example:

This command executes successfully:

```
620ZAD EV 12 HOLD ZAD EV 89 HOLD
```

This command generates an error:

```
620ZAD EV 12 HOLD 620ZAD EV 89 HOLD
```

Text Conventions

This table describes the special conventions used in this chapter to flag information in the operator command explanations. To format your operator commands correctly, it is important that you understand how these text conventions are used in this chapter.

Convention	Description
UPPERCASE	Type the word exactly as shown.
lowercase	Variable. You must supply the value.
UPPERlower	Illustrates command abbreviations. Upper case portion is required; lower case portion is optional.

Command Abbreviations

Most Zeke operator commands and parameters are abbreviated. Command and parameter abbreviations can be as few as the first two letters. The shortest form acceptable to the system for each command is illustrated in the command's syntax diagram. Parameter abbreviations are shown in the parameter description table following each command's syntax diagram.

For example, this is how the syntax for the ZMAP would be illustrated:



You can use these parameters with the ZMAP command. (Notice that the ALL parameter cannot be abbreviated, but the CLASS parameter can be abbreviated as CL.).

Parameter	Description
none	Displays only the initiators running active Zeke jobs. For example: `ZMAP`
ALL	Displays initiators running active Zeke jobs *and* non-Zeke jobs. Only jobs that started *after* Zeke came up are listed. For example: `ZMAP ALL`

Command Syntax

Enter at least one space between a command and its first parameter. For example:

```
ZHOLD SYSTEM
```

When you enter a value with a parameter, enter the value immediately following the parameter. Separate the value and parameter with either one or more spaces, or an equal sign.

For example, these two commands are equivalent:

```
ZDISPLAY JOB JOBNAME1
ZDISPLAY JOB=JOBNAME1
```

Multiple Parameters in One Command

When you enter multiple parameters with one command, separate the parameters by either one or more spaces, or by a comma, but not both.

For example, These two commands are equivalent:

```
ZDISPLAY DQ,JOB
ZDISPLAY DQ JOB
```

This is an example of a valid command that contains multiple parameters (one of which specifies a value):

```
ZD DQ,DONE,LATE,APP XYZ,JOB
```

JOB Parameter

Because some Zeke operator commands require the JOB parameter to be followed by a jobname or selection criteria, you cannot use a comma after the JOB parameter in any command. Either use spaces to separate the parameters when JOB is to be included, or enter JOB as the last parameter.

For example:

This command is valid:

```
ZD JOB LATE
```

This command is valid:

```
ZD LATE,JOB
```

This command receives a syntax error:

```
ZD JOB,LATE
```

Delimiters

For parameters that require you to enter an operand (e.g., application ID, group ID, event name, jobname, etc.), use delimiters when the operand could be confused with another parameter.

For example, if you want to display scheduled events that have an application ID of VER, use delimiters so that the operand is not confused with the VER parameter, which is used to select events based on their version number. For example:

```
ZD APPL (VER)
```

Caution! These characters *cannot* be used as delimiters:

- dollar sign ($)
- question mark (?)
- number/pound sign (#)
- at sign (@)
- asterisk (*)

Wildcard and Placeholder Characters

This section explains the rules for using wildcard and placeholder characters in selection parameters.

In Application, Group, and User ID Selection Criteria

This section explains how to use wildcards and placeholders in operands entered for these parameters:

- `APPLication`
- `GRoupid`
- `USerid`

You can specify an asterisk (`*`) as a wildcard for one or more characters. The wildcard character functions in these ways:

- At the end of an operand string, it selects any name (of any valid length) that *begins* with the specified characters. For example, this command displays all events with user IDs beginning with PAY:

  ```
  ZDISPLAY USERID PAY*
  ```

- At the beginning of an operand string, it selects any name (of any valid length) that *ends* with the specified characters. For example, this command displays all events with group IDs ending with PR:

  ```
  ZD GROUP *PR
  ```

- In the middle of an operand string, it performs a wildcard search for any name matching the specified beginning and ending characters, plus any characters in between. For example, this command displays all events with application IDs that start with PR and end with 0001:

  ```
  ZD APPL PR**0001
  ```

You can specify a question mark (?) as a placeholder for any unknown, single character. For example, this command deletes the first scheduled event that matches with the application ID beginning with P in position 1, any letter in position 2, and Y in position 3:

```
ZDEL APPL P?Y
```

You can use wildcards and placeholders in combination. For example, this command adds events with the application ID beginning with Q in position 1, any letter in position 2, and R in position 3 to the schedule:

```
ZADD APPL Q?R*
```

In Job and Event Name Selection Criteria

This section explains how to use wildcards in operands for these parameters:

- JObname
- ENAMe (event name)

You can specify an asterisk (*) as a wildcard for one or more characters. The wildcard character functions in these ways:

- At the beginning of the operand string, it can be used to select all strings starting with the operand. For example, this command displays all jobs starting with PAY:

```
ZDISPLAY JOB *PAY
```

- Anywhere else in the operand, it can be used as a placeholder for any single unknown character. For example, this command displays events with a name starting with QU followed by any two characters, and ending with R01:

```
ZDISPLAY ENAME QU**R01
```

Command Acceptance

Zeke either rejects a command with an error message, or processes the command and reports the results as command output.

For example, assume you issue the command ZD JOB LATE to display all late job events. Zeke displays all late job events. If no job events are late, Zeke responds with an appropriate message. For example:

```
Z0909W NO QUALIFYING ENTRIES.
```

When one or more events match the selection criteria, Zeke also displays the number of selected events. For example:

```
Z09141 NUMBER OF EVENTS SELECTED WAS 3.
```

Command Authentication

Zeke operator commands (prefixed with Z) are authenticated through Zeke internal security, as well as external security (if enabled). By default, commands are authenticated against profiles created in the Z$CMD class in the external security product. See the Security chapter in the *ASG-Zeke Scheduling for z/OS User's Guide* for more information on command security.

Global Parameters

Many parameters—mainly those used for event selection—are valid across all operator commands.

Global General Parameter

These global parameters are available to be used with any Zeke operator command:

Parameter	Description
IGNERRor	Controls whether command output error and warning messages can result in a non-zero command processor return code. This applies only to command processor messages, and not syntax or parser error messages (which indicate the command was not parsed successfully). **Note:** This parameter is intended as a workaround for individual commands that incorrectly cause failures due to PTF Z610ATCC. These are the valid values: NO Default. Error and warning messages *are not* ignored and can result in a non-zero return code. This option is assumed if IGNERROR is omitted. YES Command processor error and warning messages are ignored in generating the return code; regardless of error or warning messages, the return code is 0. This option is assumed if IGNERROR is specified with no value.

LOGRes	Indicates whether the command output is written to the data space audit log. These are the valid values:
	NO Command output *is not* written to the data space audit log. This is the action if the LOGRES parameter is omitted. For example, the output from these commands *is not* written to the data space audit log:
	`ZID`
	`ZID LOGRES=NO`
	YES Default. Command output is written to the data space audit log. For example, the output from these commands is written to the data space audit log:
	`ZID LOGRES`
	`ZID LOGRES=YES`

Global Event Selection Parameters

Many global event selection parameters are available to be used with any Zeke operator command that requires event selection (i.e., ZALTER, ZDISABLE, ZDISPLAY, ZENABLE, ZHOLD, ZOK, ZREFRESH, ZRELEASE, ZSTATUS, and ZSCAN).

For each operator command that requires event selection, you must include at least one selection parameter (either global or specific to that command). You must either specify criteria for selecting the events to be affected or indicate the operation be performed for all events. (If you enter this type command with no additional event selection parameters, no events are selected and Zeke does not execute the request.)

Generally, Zeke searches scheduled events that have not been dispatched (in schedule time sequence). Zeke does not include completed (i.e., Done, Active, Pending, Failed, Forced Done, etc.) or disabled events in the search. For some commands (e.g., ZDELETE), Zeke executes the specified command only for the first event that matches the selection criteria. To enable Zeke to search and/or select all scheduled events (including completed or disabled events), you must include the global event selection parameter ALL (described in this section).

This table lists the global event selection parameters and notes instances where the function of a specific parameter might differ among commands:

Parameter	Description

`ALL`	When you use the ALL parameter, Zeke searches all scheduled events (regardless of their status), and then executes the specified command for all events that match the selection criteria.
	Without the ALL parameter, Zeke searches scheduled events that have not been dispatched (in schedule time sequence) and then executes the specified command only for the first event that matches the selection criteria.
	Examples:
	This command displays all scheduled events (regardless of event status):
	`ZDISPLAY ALL`
	This command places an operator hold on all scheduled events:
	`ZHOLD ALL`
	This command selects nothing and does not execute a hold:
	`ZHOLD`

`APplication`	Selects events with the specified application ID (up to eight characters long). **Note:** You can use wildcard and placeholder characters in your selection criteria. See "Wildcard and Placeholder Characters" on page 259.

Parameter	Description

	Examples:
	This command displays events that have an application ID beginning with P:
	`ZD APP P*`
	This command adds events that have an application ID beginning with QU:
	`ZADD APPL QU*`
	This command adds events that have an event name beginning with JOB and an application ID beginning with ACCT:
	`ZADD ENAME *JOB APPL ACCT*`
	Note:
	For ZADD, selection is controlled by the MultAp generation option (see page 558).

`CLass`	Specifies the class specification for a job event. Zeke selects scheduled job events that have the specified class in their class lists.
	With the ZDISPLAY command, you can specify an asterisk (*) to display all events that have a class specification.
	Examples:
	This command selects all events that have a class specification:
	`ZD CLASS *`
	This command selects events that have class A as a class specification:
	`ZD CLASS A`
	This command selects events in the dispatch queue that have any class specified in their class lists:
	`ZD DQ CL *`
	This command selects events in the dispatch queue that have class A as a class specification:
	`ZD DQ CL A`

`DAte`	Specifies a Julian date (in *yyddd* or *yyyyddd* format). Zeke selects scheduled events that have this schedule date. For example, this command selects the SQR for event 27 for January 1, 2020: `ZD EV 27 DA 2020001` **Note:** When using *yyddd* format, if *yy* is greater than or equal to `50`, it is considered a twentieth century date (`19`*xx*); if it is less than `50`, it is considered a twenty-first century date (`20`*xx*).

Parameter	Description
	With many commands that use global event selection parameters, you can specify a date value of `99999`. Zeke selects the events that match the criteria and have the earliest schedule date. These commands enable you to specify `DATE 99999`: ZALTER ZDELETE ZDISABLE ZDISPLAY ZENABLE ZHOLD ZOK ZREFRESH ZRELEASE ZSCAN For example, this command selects all uncompleted events in the schedule that have an application ID of `JEM`. Zeke disables the events with the earliest schedule date: `ZDISABLE APP JEM DATE 99999` Specifying `DATE 99999` selects the earliest schedule date across all the events that match the selection criteria (without considering the event number). In the example, assume the first matching event has the schedule dates 2017001 and 2017002 and the next matching event has the schedule dates 2017002 and 2017003. In this case, Zeke uses the earliest date, 2017001, when processing the ZDISABLE command. When determining the earliest date, Zeke ignores events that have any of these statuses: • Done (including events set to DONE through a ZALTER command) • Pending • Active • ABOK • Set to WHENOK or RUN through a ZALTER command

Parameter	Description

	When altering individual WHEN conditions to 'satisfied,' Zeke does not consider whether the condition already is satisfied when determining the earliest date.
	
	For example, assume that event 1 has the schedule dates 2017001 and 2017002 and has the WHEN condition `EOJ JOBA`. The condition already is satisfied for schedule date 2017001. If you issue this operator command:
	
	`ZALTER EVENT 1 WHENOK EOJ JOBA DATE 99999`
	
	then the command will select 2017001 as the earliest date even though the command will have no effect on the schedule record for that date.
	
	Note:
	When the `DATE 99999` parameter is used, Zeke ignores some selection criteria until the earliest date has been determined:
	
	For ZALTER, Zeke considers these criteria *after* the date is determined:
	
	`TIMEOK`
	`NOTDUROK`
	`RERUN`
	`NORERUN`
	`TRIG`
	`NOTRIG`
	
	For ZENABLE, Zeke does not consider whether the event is disabled.
	
	For ZHOLD, Zeke does not consider whether the event is on hold.
	
	For ZOK, Zeke does not consider whether the event is waiting on an operator approval.
	
	For ZREFRESH, the `DATE 99999` parameter is not valid since completed events are ignored.
	
	For ZRELEASE, Zeke does not consider whether the event is on hold.
	
	For ZSCAN, Zeke does not consider whether the event has JESQ JCL.

Parameter	Description
	When using the DATE parameter in a Zeke operator commands or a system command issued through an SCOM event, ZEKESET batch job, or ZCOM event, you can specify the keyword SCHDT in place of a Julian date. Zeke substitutes the command event's schedule date for the SCHDT value before issuing the command. For example: `ZADD JOB JOBADD DATE SCHDT` **Notes:** • For the SCHDT keyword to be substituted in a ZEKESET batch job, it must appear in a SET SCOM or SET ZCOM statement. • The SCHDT keyword works differently in a ZEKESET batch job submitted outside of Zeke. In this case, the current system date is substituted in place of SCHDT.

DOne	Selects events that have been completed successfully or disabled.
	If you omit this parameter, then Zeke does not select completed or disabled events unless you specify the EVENT parameter.
	Examples:
	This command selects all completed events:
	`ZD DONE`
	This command selects all completed job events:
	`ZD DONE,JO`

DQ	Selects events that are in the dispatch queue.
	If you want to select events that are in the schedule, but not in the dispatch queue yet, specify SQ.
	If you do not specify either parameter, both are assumed.
	Examples:
	This command selects all events that are currently in the dispatch queue:
	`ZD DQ`
	This command selects only Zeke command events in the dispatch queue.
	`ZD DQ,ZC`

ENAMe	Selects events with the specified event name (up to 12 characters long). **Note:** You can use wildcard and placeholder characters in your selection criteria. See "Wildcard and Placeholder Characters" on page 259.

Parameter	Description

	Examples:
	These commands select events with names that begin with PAY:
	`ZALTER ENAME *PAY EARLY 1000`
	`ZDISPLAY ENAME *PAY`
	This command selects events with names that have any characters in positions 3 through 12:
	`ZDISPLAY ENAME 01**********`
	Note:
	For ZADD, selection is controlled by the MultEn generation option (see page 558).

`EVent`	Specifies one or more event numbers.
	Enclose multiple event numbers in parentheses and separate them with commas or spaces. You can specify up to 20 event numbers.
	Examples:
	This command selects event 143 only:
	`ZD EV 143`
	This command selects events 10, 11, and 12:
	`ZA EV (10,11,12) WHENOK`
	This command selects events 14, 20, and 50:
	`ZADD EV (14 20 50)`

`FRom`	Specifies a time, in *hh*:*mm* format. Selects events scheduled after this time. For example, this command selects job that are scheduled after 12:00 P.M: `ZD JOB FRO=1200`
`GE`	(Greater than or Equal) Selects events scheduled on or after the specified date (in *yyyyddd* format). This parameter must follow the keyword DATE. For example, this command selects all events scheduled on or after January 1, 2020: `ZD DATE GE 2020001`
`GRoupid`	Selects events with the specified group ID (up to three characters long). **Note:** You can use wildcard and placeholder characters in your selection criteria. See "Wildcard and Placeholder Characters" on page 259.

Parameter	Description

	Examples:
	This command selects events with a group ID beginning with 1 (depending on the setting of the MultGr generation option):
	`ZADD GROUP 1*`
	This command selects only events with a group ID of 011:
	`ZD GROUP 011`
	This command selects any events that have a group ID that begins with T:
	`ZD GROUP T*`
	Use delimiters when the value could be confused with a parameter. For example:
	`ZD GROUP (SYS)`
	Note:
	For ZADD, selection is controlled by the MultGr generation option (see page 559).

`HOld`	Selects an event on hold, or with the specific hold specified.
	Examples:
	This command displays all jobs on hold:
	`ZD HOLD`
	This command displays all jobs on hold with a network hold:
	`ZD HOLD=NET`

`JOb`	Up to 30 characters; case is not considered. Due to space restrictions, all 30 characters do not display in any output produced. For ZDISPLAY—If entered without an operand, the JOB parameter specifies to display information for all *job* events. For example, this command displays job events with the group ID PAYROLL: `ZD GR PAYROLL JOB` **Note:** You can specify more than one event type. If you omit the event type, all types are selected. If an operand is included, the JOB parameter indicates to display job events with the specified jobname.

Parameter	Description

	Examples:
	This command displays job events that begin with PR:
	`ZD JO PR*`
	This command displays all jobs that start with PR and end with 0001:
	`ZD JOB PR**0001`
	This command displays all jobs that have P in position 1, U in position 3, 0 in position 5, 1 in position 7:
	`ZD JOB P*U*0*1*`
	For all other commands—Specifies the jobname of the event to be selected.
	Note:
	You can use wildcard and placeholder characters in your selection criteria. See "Wildcard and Placeholder Characters" on page 259.
	Examples:
	This command holds all jobs:
	`ZH JOB *`
	This command alters all job events to run class B:
	`ZA JOB * NEWCLASS B`
	Note:
	For ZADD, selection is controlled by the MultJn generation option (see page 559).

`LE`	(Less than or Equal) Selects events scheduled on or before the specified date (in *yyyyddd* format). This parameter must follow the keyword DATE. For example, this command selects all events that are scheduled on or before January 1, 2020: `ZD DATE LE 2020001`
`MSg`	Selects all message events. For example, this command displays message events: `ZD MSG` **Note:** You can specify more than one event type. If you omit the event type, all types are selected.
`NETRegid`	Selects all events waiting for a job on the specified Netregid. For example, this command displays events waiting for a job on SYSC: `ZD AT SYSC`

Parameter	Description
REXx	Selects REXX events. For example, this command selects all REXX events in the schedule: `ZD REX` **Note:** You can specify more than one event type. If you omit the event type, all types are selected.
SCHENV	Selects events with the specified scheduling environment (up to 16 characters long) requirement.
SCom	Selects all system command events. For example, this command displays all system command events in the schedule: `ZD SCO` **Note:** You can specify more than one event type. If you omit the event type, all types are selected.

SQ	Selects events that are in the schedule queue (i.e., schedule, but not yet in the dispatch queue).
	If you want to select events that in the dispatch queue, specify `DQ`.
	If you do not specify either parameter, both are assumed.
	Examples:
	This command selects job events that are currently in the schedule queue:
	`ZD SQ,JOB`
	This command selects all events that are in the schedule queue:
	`ZD SQ`

SYstem	Selects events owned by the specified system. If this parameter is omitted, events are selected only from the system where the command was entered. If the specified system is not the same system from which the command is issued, only events in the schedule queue are selected. With the ZDISPLAY command, include an asterisk (*) to display information from all systems that share the same Zeke database.

Parameter	Description

	Examples:
	This command alters an event on System B:
	`ZA EV 5 WHENOK SYS B`
	This command displays the active SQRs from all systems:
	`ZD SYSTEM *`
	This command displays successfully completed events from all systems:
	`ZD DONE SYS *`
	Note:
	This parameter is ignored when used with the ZHOLD and ZRELEASE operator commands.

`TARget`	Selects events in the schedule with the specified target. For example, this command selects all events with a target value of SYS2: `ZD TARG=SYS2`

`TO`	Selects events scheduled before the specified time (in *hh*:*mm* format).
	Examples:
	This command selects job events that are scheduled before noon:
	`ZD JOB TO 1200`
	This command selects all events scheduled between noon and 1600 hours (4:00 P.M.):
	`ZD FRO=1200 TO=1600`

`USerid`	Specifies the user ID to match.
	Note:
	You can use wildcard and placeholder characters in your selection criteria. See "Wildcard and Placeholder Characters" on page 259.
	Examples:
	This command selects only events with user ID of OPERATOR:
	`ZD USERID OPERATOR`
	This command selects any event that has a user ID that begins with OP:
	`ZD USERID OP*`
	This command selects any event that has a user ID that begins with ACC:
	`ZD USER (ACC)`

Parameter	Description
	Note: For ZADD, selection is controlled by the MultUs generation option (see page 560).

VCom	Selects only VM command events.
	Examples:
	This command selects all VM command events in the schedule:
	`ZD VCOM`
	This command selects all VM command and Zeke command events in the schedule:
	`ZD VC ZC`
	Note:
	You can specify more than one event type. If you omit the event type, all types are selected.

VERsion	Selects the specified version of the event. To specify a specific version, enter version number (up to five digits long).
	Examples:
	This command deletes all SQRs for version 2 of event 30:
	`ZDEL EV 30 VER 2`
	This command deletes only the SQR for schedule date January 1, 2020 for version 2 of event 30:
	`ZDEL EV 30 DATE 2020001 VER 2`
	If a version number is not specified, all versions of the event are selected.
	Examples:
	This command deletes all SQRs for event 30:
	`ZDEL EV 30`
	This command deletes all versions of event 30 with a matching schedule date of January 1, 2020:
	`ZDEL EV 30 DATE 2020001`

WORKCenter	Selects only work center events. For example, this command selects all work centers in the schedule: `ZD WORKC`

Parameter	Description
	Note: The ZD WORKC command is effective only if the LoadComm generation option is set to Y (see page 554). **Note:** You can specify more than one event type. If you omit the event type, all types are selected.

ZCom	Selects only Zeke command events.
	
	Examples:
	
	This command selects all Zeke command events in the schedule:
	
	`ZD ZCOM`
	
	This command selects all Zeke command and VM CP command events in the schedule:
	
	`ZD ZC VC`
	
	Note:
	You can specify more than one event type. If you omit the event type, all types are selected.

ZADD Command

The ZADD command creates a schedule queue record (SQR) based on an existing event master record (EMR). You can use this command to add any type of event (including an on-request event) to the schedule.

The ZADD command is the only method for scheduling permanent events (which run an unlimited number of times).

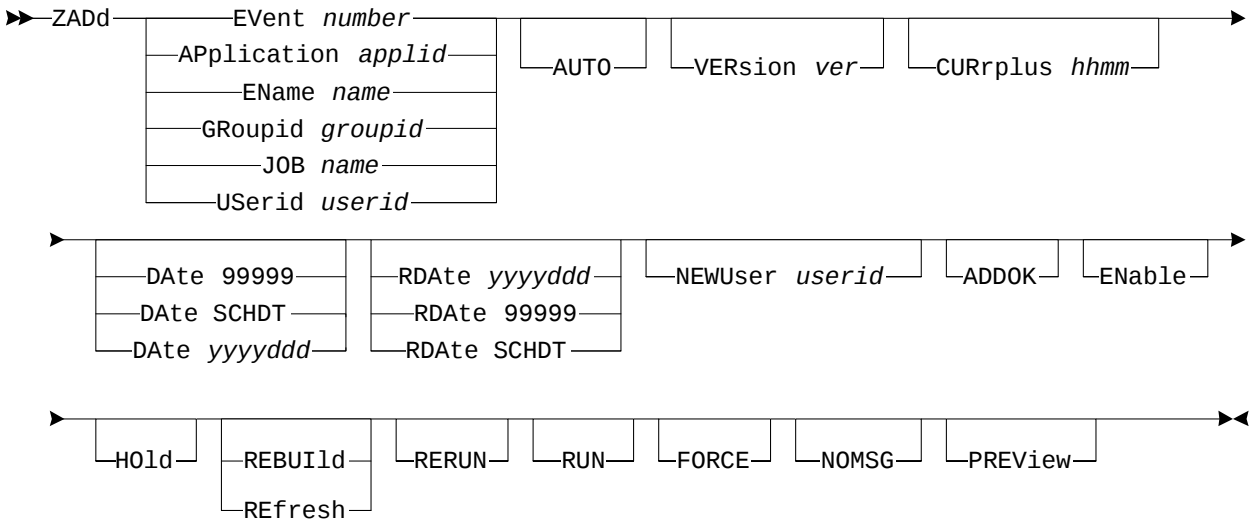
Note:

When you need to add multiple events, consider any predecessor/successor relationships before you submit the request. If you are attempting to add multiple events that have predecessor/successor relationships, you must use a single ZADD command to ensure that all of the events are added to the schedule before Zeke dispatches any of them. If you specify related events on separate ZADD commands, a predecessor event could be completed before a successor has been added to the schedule (resulting in missed triggers).

Caution! To prevent adding a large number of events unintentionally, include the PREVIEW parameter.

See the *ASG-Zeke Scheduling for z/OS User's Guide* for information on manually adding events to the schedule.

Syntax



Parameters

The ZADD command requires either the *event number* as a selection parameter or *one or more* of these selection parameters:

- Application ID
- Event name
- Group ID
- Jobname
- User ID

All other parameters are optional.

Note:

If you include the EVENT parameter, these same parameters are ignored.

You can use these parameters with the ZADD command:

Parameter	Description
ADDOK	Adds the specified event to the schedule, but requires the event to wait for an operator OK before it can be dispatched. For example: `ZADD EV 500 ADDOK`
APplication	Adds events with the specified application ID (up to eight characters long) to the schedule. **Note:** You can use wildcard and placeholder characters in your selection criteria. See "Wildcard and Placeholder Characters" on page 259. The MultAp generation option (see page 558) controls the action to take when more than one event matches the specified application ID (i.e., whether to add only the first one only, none, or all). **Note:** If you include the EVENT parameter, the APPL parameter is ignored.

Parameter	Description

	Examples:
	This command adds events with the application ID of ACCT to the schedule (depending on the MultAp setting):
	`ZADD APPL ACCT`
	This command adds events with the application ID beginning with Q in position 1, any letter in position 2, and R in position 3 to the schedule (depending on the MultAp setting):
	`ZADD APPL Q?R*`
	This command adds events with the application ID beginning with QU to the schedule (depending on the MultAp setting):
	`ZADD APPL QU*`
	This command adds events with the event name beginning with JOB and an application ID beginning with ACCT to the schedule (depending on the MultAp and MultEn settings):
	`ZADD ENAME *JOB APPL ACCT`
	Note:
	If you specify multiple Mult*xx*-controlled parameters, and if Zeke finds multiple event matches, the parameter with the most restrictive Mult*xx* setting overrides the other parameters. For example:
	If MultEn is set to `F` (first) and MultAp is set to `A` (all) and Zeke finds multiple matches for this command:
	`'ZADD ENAME *JOB APPL ACCT',`
	Then, MultEn overrides MultAp so that Zeke adds only the first match.

`AUTO`	For an event that has not completed (has an Active or Failed status, or has a Times value greater than `1`), this parameter adds 1 to the number of dispatch times. For an event that has completed (has a Success status), the REFRESH and ENABLE parameters are assumed. For example: `ZADD EV 19 AUTO` **Note:** Increasing the count causes the event to run the indicated number of times in this schedule run, but does not make the SQR a recurring event (which must be set up in the EMR). For a recurring event, Zeke resets the WHEN conditions at dispatch time, and the event can become WHEN-satisfied even while it is active.

Parameter	Description
	Note: This parameter is not valid for a work center event or permanent event.
CURrplus	Updates the scheduled time on the added or refreshed event by adding the time specified in this parameter to the event's current Sched time. If the Sched time is zero, the specified time is added to the current system time and saved in the Sched time field. Enter the amount of time you want to add to the current Sched time or system time to create the new Sched time for the event. You specify the hour and minutes (in *hhmm* format). The valid values range from `0000` through `4800`. For example, this command changes the start time of event 123 to one hour and 30 minutes from the current system time: `ZADD EVENT 123 CURRPLUS 0130` If the resulting Sched time value is greater than `4800` (48 hours), Zeke issues message Z09B4E and does not add the job to the schedule. Zeke also considers the specified RDATE (run date) when recalculating the Sched time. • If the run date is yesterday, then the new Sched time is adjusted by 24 hours. • If the run date is specified and is prior to yesterday or after the current date, Zeke issues message Z09B4E and does not add the job to the schedule.
DAte	Specifies the schedule date (in *yyyyddd* or *yyddd* format) for the event if it is different from the system date. **Note:** (When using *yyddd* format, if *yy* is greater than or equal to `50`, it is considered a twentieth century date (`19`*xx*); if it is less than `50`, it is considered a twenty-first century date (`20`*xx*). For example, this command adds event 100 to the schedule with a schedule date of January 1, 2020: `ZADD EV 100 DA 2020001` By issuing the ZADD command with a date value of `99999`, you can add an event with the schedule date of the earliest event in the schedule that is dependent on the event being added.

Parameter	Description
	For example, this command searches the schedule for the earliest event (the event with the earliest schedule date) that has not been dispatched because it is waiting on event 234. Zeke then adds event 234 is then added with that schedule date. If no events are waiting on event 234, then Zeke adds event 234 with the current system date as the schedule date: `ZADD EV 234 DATE 99999` If you issue a single ZADD command to add multiple events, a schedule date is calculated for each event as it is added. If the ZADD SCHDT command is issued from either a ZCOM event, a Zeke operator (`Z`) command or system (`C`) command in an SCOM event, or a SET SCOM or SET ZCOM statement from a ZEKESET batch job dispatched as a Zeke event, Zeke adds the new event to the schedule with the same schedule date as the event that is used to issue the command. If the ZADD SCHDT command is issued from any other environment, the current system date is used as the new event's schedule date.
`ENABLE`	Updates the status of the SQR to enabled (if it is disabled). For example: `ZADD EV 19 ENABLE`
`ENAMe`	Adds events with the specified event name (up to 12 characters long) to the schedule. **Note:** You can use wildcard and placeholder characters in your selection criteria. See "Wildcard and Placeholder Characters" on page 259. The MultEn generation option (see page 558) indicates the action to take when more than one event matches the specified name (i.e., it indicates whether to add the first one only, add none, or add all). **Note:** If the operand in fewer than 12 characters, the comparison is made only for the number of characters entered (including wildcard characters). **Note:** If the EVENT parameter is included, the ENAME parameter is ignored. Examples: This command adds event name TESTJOB00001 to the schedule (depending on the MultEn setting): `ZADD ENAME TESTJOB00001`

Parameter	Description

	This command adds event names beginning with the character string EVNTNAM and any alphanumeric value in position 8 to the schedule (depending on the MultEn setting):
	`ZADD ENAME EVNTNAM?`
	This command adds eight-character event names beginning with 01 to the schedule (depending on the MultEn setting):
	`ZADD ENAME 01******`
	This command adds 12-character event names beginning with 01 to the schedule (depending on the MultEn setting):
	`ZADD ENAME 01**********`
	This command adds events with the event name beginning with JOB and an application ID beginning with ACCT to the schedule (depending on the MultEn and MultAp settings):
	`ZADD ENAME *JOB APPL ACCT`
	Note:
	If you specify multiple Mult*xx*-controlled parameters, and if Zeke finds multiple event matches, the parameter with the most restrictive Mult*xx* setting overrides the other parameters. For example:
	If MultEn is set to F (first) and MultAp is set to A (all) and Zeke finds multiple matches for this command:
	`'ZADD ENAME *JOB APPL ACCT',`
	Then, MultEn overrides MultAp so that Zeke adds only the first match.

`EVent`	Adds events with the specified event numbers to the schedule. Follow this parameter with one or more event numbers. Enclose multiple event numbers in parentheses and separate them with commas or spaces.
	Note:
	When any of the events to be added have predecessor/successor relationships, be sure to add them using a single ZADD command.
	Up to 20 event numbers can be entered. For example:
	`ZADD EV 19`
	`ZADD EV (19,45,100)`
	Note:
	If this parameter is used, any included APPLICATION, ENAME, GROUP, JOB, and USERID parameters are ignored.

Parameter	Description
FORCE	Releases an event's acquired resources before the event is re-added using the ZADD REBUILD command. This produces the same result as issuing the ZRESOURCE RELEASE command to release acquired resources followed by the ZADD REBUILD command. FORCE does not release kept resources. If there are no resources for the event, this parameter is ignored. For example: `ZADD EV 12 REBUILD FORCE`

GRoupid	Adds events with the specified group ID to the schedule.
	The MultGr generation option (see page 559) indicates the action to take when more than one event matches the specified group ID (i.e., it indicates whether to add the first one only, add none, or add all). For example, this command adds events with the group ID of OPR to the schedule (depending on the MultGr setting):
	`ZADD GROUP OPR`
	Note:
	If the EVENT parameter is included, the GROUP parameter is ignored.
	Note:
	If you specify multiple Mult*xx*-controlled parameters, and if Zeke finds multiple event matches, the parameter with the most restrictive Mult*xx* setting overrides the other parameters. For example:
	If MultEn is set to F (first) and MultAp is set to A (all) and Zeke finds multiple matches for this command:
	`'ZADD ENAME *JOB APPL ACCT',`
	Then, MultEn overrides MultAp so that Zeke adds only the first match.

HOld	Places an operator hold on the event after it is added, refreshed, or enabled. **Note:** This parameter is not valid for work center events.

Parameter	Description

JOB	Adds events with the specified jobname to the schedule. Mixed case jobnames are not allowed.
	
	Note:
	You can use wildcard and placeholder characters in your selection criteria. See "Wildcard and Placeholder Characters" on page 259.
	
	The MultJn generation option (see page 559) indicates the action to take when more than one event matches the specified jobname (i.e., it indicates whether to add the first one only, add none, or add all).
	
	For example, this command adds events with the jobname of TESTJOB to the schedule (depending on the MultJn setting):
	
	`ZADD JOB TESTJOB`
	
	Note:
	If the EVENT parameter is included, the JOB parameter is ignored.
	
	Note:
	If you specify multiple Mult*xx*-controlled parameters, and if Zeke finds multiple event matches, the parameter with the most restrictive Mult*xx* setting overrides the other parameters. For example:
	
	If MultEn is set to F (first) and MultAp is set to A (all) and Zeke finds multiple matches for this command:
	
	`'ZADD ENAME *JOB APPL ACCT',`
	
	Then, MultEn overrides MultAp so that Zeke adds only the first match.

NEWUser	The new user ID to be associated with the SQR being created or rebuilt. If this parameter is specified, the security call checks this new user ID for ALTER level access to the SQR. If this parameter is not specified, the security call checks the user ID specified in the EMR. After the SQR is created or rebuilt with a new user ID, all subsequent security calls will use the NEWUSER value (in the Userid field in the SQR). This parameter allows different versions of an SQR to be associated with different user IDs. For example, this command adds a version 33 SQR to the schedule with a user ID of ABC for each of the specified events: `ZADD EV (5,6,7) VER 33 NEWUSER ABC`
NOMSG	Suppresses messages Z0905I and Z0906W.

Parameter	Description
`PREView`	Produces a list of events that would be added to the schedule by the command string (without executing it). For example, this command displays a preview of events to be added from group `OPR`: `ZADD GROUP OPR PREVIEW`
`RDate`	Specifies the run date to be associated with the SQR (in *`yyyyddd`* or *`yyddd`* format). An event that has a future run date is not run until that date; however, the event's WHEN conditions may or may not be tracked throughout the period depending upon the setting of the generation option Trigdt. **Note:** When using the *`yyddd`* format, if *`yy`* is greater than or equal to `50`, it is considered a twentieth century date (`19`*`xx`*); if it is less than `50`, it is considered a twenty-first century date (`20`*`xx`*). By issuing the ZADD command with an RDATE value of SCHDT, a new event can be added to the schedule with the same run date as the event issuing the ZADD command. If the ZADD SCHDT command is issued from either a ZCOM event, a console or 'Z' command in an SCOM event, or a SET SCOM or SET ZCOM statement from a ZEKESET batch job dispatched as a Zeke event, the new event is added to the schedule with the same run date as the event issuing the command. If the ZADD SCHDT command is issued from any other environment, the current system date is used as the new event's run date. By issuing the ZADD DATE command with a date value of `99999`, you can add an event with the run date of the earliest event in the schedule that is dependent on the event being added. For example, this command searches the schedule for the event with the earliest run date that has not been dispatched because it is waiting on event 234. Zeke adds event 234 with that run date. If no jobs are waiting on event 234, Zeke adds it with the current system date as the run date: `ZADD EVENT 234 RDATE 99999` If you issue a single ZADD command to add multiple events, a run date is calculated for each event as it is added.

Parameter	Description
REBUIld	Re-creates the SQR from its EMR. (If the SQR does not exist, this parameter is ignored.) This produces the same result as deleting an SQR and re-adding it. • Resets all WHEN conditions • Reflects any EMR changes • Resets any ZALTER changes to the EMR values For example: `ZADD EV 19 REBUILD`
REfresh	Refreshes an SQR (regardless of its status). For example: `ZADD EV 19 REFRESH` **Note:** ZADD REFRESH does not place an operator hold on the event like ZREFRESH does. You must include the HOLD parameter to hold the refreshed SQR.
RERUN	Adds the RERUN designation to the SQR. The event will run according to the attributes of the active SQR (the event is not rebuilt from the EMR). The value of the TrigRrn generation option determines whether the event will trigger the WHEN conditions of other events. Use the ZALTER NORERUN command to remove the RERUN designation.
RESTART	Refreshes a failed SQR and flags it to be restarted at the failed/canceled step. **Note:** If you specify to restart a refreshed event, the same system ID used for the failed execution is used for this execution (even if a pool ID was specified in the EMR).
RUN	Adds a new SQR to the schedule and satisfies these conditions automatically so that it is ready to run: • TIMEOK • WHENOK • OPEROK • NOTDURING • ZRELEASE

Parameter	Description
	For example: `ZADD JOB TSO1H001 RUN` You can include the HOLD parameter to place the event on hold after it is added to the schedule. The event is ready to run when it is released. These conditions are checked before Zeke dispatches the event: • SYSTEM HOLD • INITIATOR • TAPES • RESOURCES NOTDURING clauses are ignored.

`USerid`	Adds events with the specified user ID to the schedule.
	The MultUs generation option (see page 560) indicates the action to take when more than one event matches the specified user ID (i.e., it indicates whether to add the first one only, add none, or add all).
	For example, this command adds events with the user ID of DEV to the schedule (depending on the MultUs setting):
	`ZADD USER DEV`
	Note:
	If the EVENT parameter is included, the USERID parameter is ignored.
	Note:
	If you specify multiple Mult*xx*-controlled parameters and Zeke finds multiple matching events, the parameter with the most restrictive Mult*xx* setting overrides the other parameters. For example, if MultEn is set to F (first) and MultAp is set to A (all), and Zeke finds multiple matches for this command, then MultEn overrides MultAp so that Zeke adds only the first match:
	`'ZADD ENAME *JOB APPL ACCT',`

`VERsion`	Adds the specified version of the event to the schedule. However, if the Verload field on the EMR for the event is set to zero, you can only add a version zero SQR. The maximum allowed versions of an event is 32,767. **Note:** ASG recommends running no more than 1000 versions of a single event.

Parameter	Description

	For example, this command creates an SQR for version 99 of event 5:
	`ZADD EV 5 VER 99`
	This command creates SQRs for version 99 of events 5, 6, and 7:
	`ZADD EV (5,6,7) VER 99`
	
	If you issue the ZADD command without the VER parameter, and if the Verload field is greater than zero, Zeke will add 1 to the highest version in the schedule and create an SQR for that version. For example, if there are no versions of the event in the schedule, version 1 is added. If the highest version in the schedule is 7, version 8 is added.

ZALTER Command

The ZALTER command temporarily alters SQR information, some system-related features, and initiator availability. When changing event-related information, you must include at least one of the global event selection parameters. See "Global Parameters" on page 261 for a listing.

Altering SQR information does not modify the EMR. The information is updated only for a specific occurrence of an event; its permanent definition is not altered.

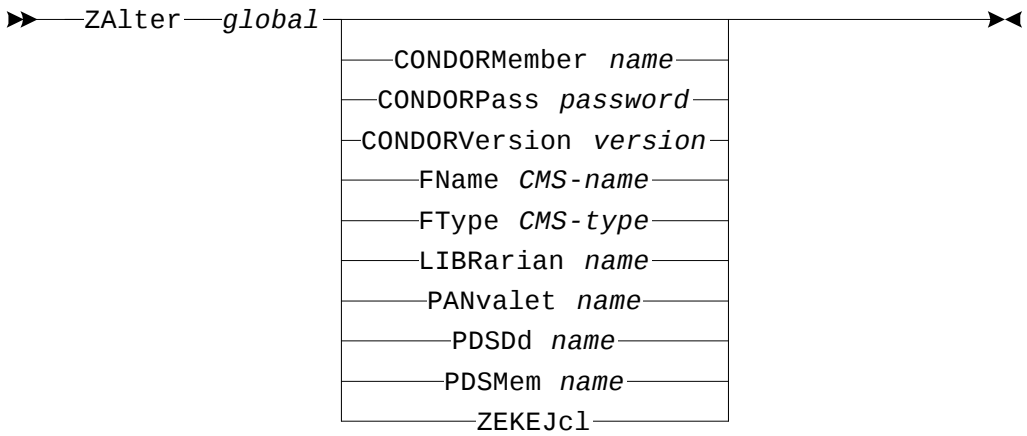
Altering JCL Source

You can alter JCL source for scheduled job events.

Note:

You cannot alter JCL source for a job that has been downloaded to Zeke Agent.

Syntax



Parameters

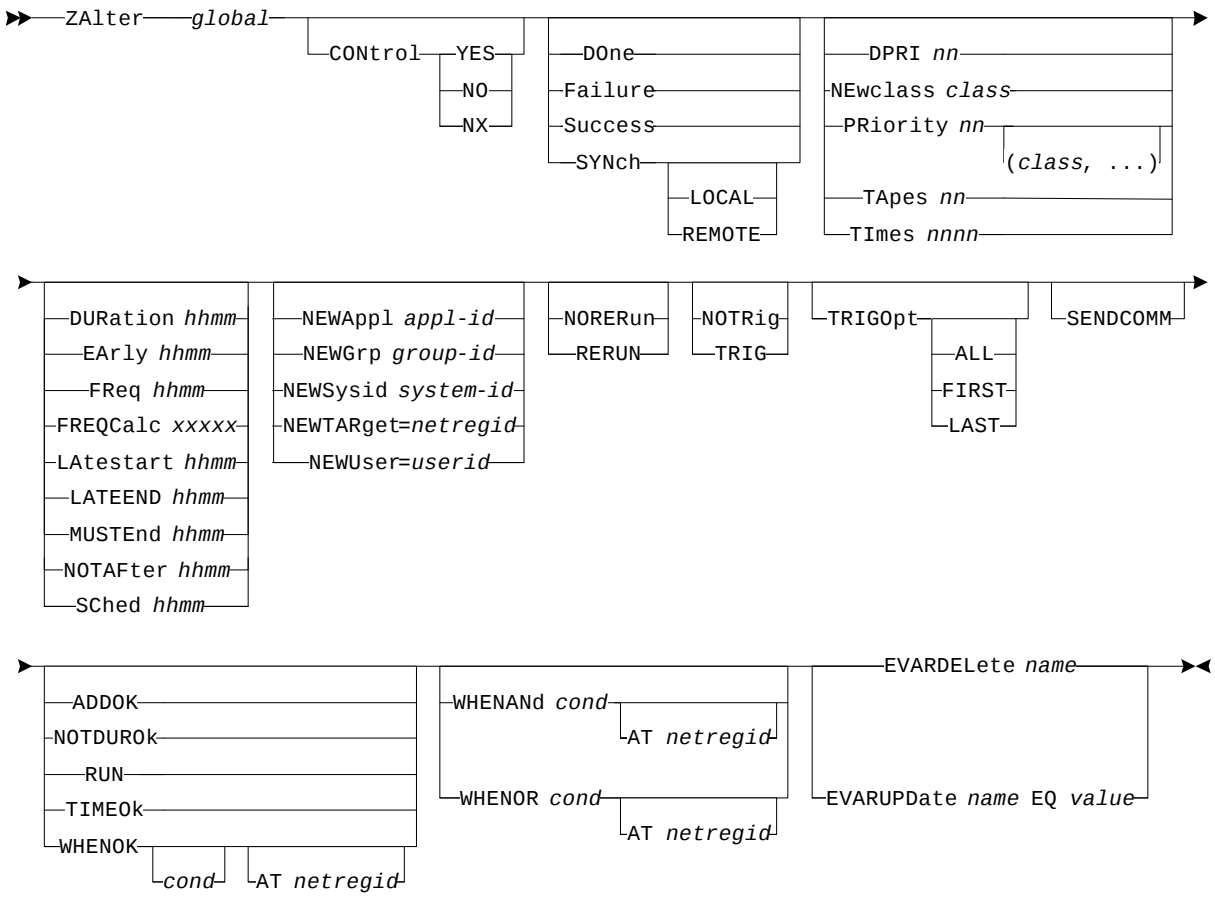
You can use these parameters with the ZALTER command:

Parameter	Description
`CONDORMember`	Condor member name that is retrieved when the event is dispatched.
`CONDORPass`	Condor password that is passed to CONDOR when the event is dispatched.
`CONDORVersion`	Condor version that is passed to Condor when the event is dispatched.
`FName`	Specifies the CMS file name that is retrieved when the event is dispatched. The filename can be up to eight characters long.
`FType`	Specifies the CMS file type that is retrieved when the event is dispatched.The file type can be up to four characters long.
`LIBRarian`	CA Librarian member name accessed during job submittal.
`PANvalet`	Panvalet member name used during job submittal.
`PDSDd`	PDS ddname used during job submittal. `ZA EV 14 PDSD ZEKDD1` Changes the PDS name for event 14.
`PDSMem`	PDS member name used during job submittal. `ZA EV 14 PDSM JOBABC` Changes the PDS member for event 14.
`ZEKEJcl`	Zeke JCL.

Altering Other Event Fields

These parameters are used with the ZALTER command to change selected SQRs. The event selection logic notifies the operator of how many records were modified.

Syntax



Parameters

At least one of the global event selection parameters (starting on page 261) is required.

Parameter	Description
ADDOK	Requires the event to wait for an operator OK before it can be dispatched. For example: `ZA EV 500 ADDOK`

Parameter	Description

AT	Valid with the WHENAND, WHENOR, or WHENOK parameter and used to add or manually satisfy ('WHENOK') a remote WHEN condition. The AT parameter specifies the Netregid of the remote system.
	Examples:
	`ZALTER EV 1 WHENAND (EOJ JOBC AT SYSB)`
	`ZALTER EV 1 WHENOR (EOJ JOBC AT SYSB)`
	`ZALTER EV 1 WHENOK (EOJ JOBB AT SYSB)`

CONtrol	Indicates whether this job is tracked as an Zeke-controlled job. Zeke-controlled jobs are tracked throughout the entire execution. These are the valid values: `YES` Default. Zeke recognizes this event as an Zeke-controlled event. `NO` Zeke does not recognize this event as an Zeke-controlled event and marks the event as SUCC upon dispatch. `NX` Zeke recognizes the event as a non-executable Zeke-controlled event. For details on non-executable events, see the *ASG-Zeke Scheduling for z/OS User's Guide*.
DOne	Forces an event to F/S (Forced Success) status. F/S satisfies the WHEN conditions for any events that reference that event. For example: `ZA JOB TS01H001 DONE`
DPRI	Updates the event's dispatch priority number. The valid values range `01` (highest) through `99` (lowest).
DURation	Updates the event's average duration time (in *hhmm* format). The valid values range from `0000` through `4759`. For example: `ZA EV 100 DUR 0300`
EArly	Updates the earliest time (in *hhmm* format) the event can be dispatched. The valid values range from `0000` through `4759`. For example: `ZA EV 14 EA 0100`
EVARDELete	Deletes the specified event variable from the event(s).
EVARUPDate	Updates the specified event variable from the event(s). If the selected event(s) does not contain the specified variable, the variable is added to the event(s).
Failure	Marks the event with an F/F (i.e., Forced Failure) status and triggers any failure dependencies. For example: `ZA EV 35 FAILURE`

Parameter	Description
FReq	Updates the time (in *hhmm* format) between dispatches for a recurring event. The valid values range from 0000 through 4759. For example: `ZA JOB TS01H001 FR 0010`
FREQCalc	Valid for recurring events only. Indicates how to calculate the next dispatch time. These are the valid values: **SCHED** Reschedule the event by the actual schedule time, no matter when the event was dispatched or completed. For example, suppose an event is scheduled to run at 10:00, and then two more times (every two hours). The second and third events will remain scheduled for 12:00 and 2:00 regardless of the completion time of the first event. **CLOCK** Reschedule the events based on the completion time of the first occurrence. For example, suppose an event is scheduled to run at 10:00, and then two more times (every two hours). If the event is completed at 10:15, then the second event is scheduled for 12:15. Then, when the second event is completed, the third event is scheduled for two hours later.
LATEEND	Updates the event's 'late end' time (in *hhmm* format). The valid values range from 0000 through 4759. For example: `ZA JOB *PAY LATEEND 2605`
LAtestart	Updates the event's 'late start' time (in *hhmm* format). The valid values range from 0000 through 4759. For example: `ZA JOB *PAY LA 1005`
MUSTEnd	Updates the event's 'must end' time (in *hhmm* format). The valid values range from 0000 through 4759. For example: `ZA EV 1 MUSTE 1105`
NEWAppl	Updates the event's application ID.

NEwclass	Updates the class list for a job event. If one or more values are entered, the event can only run as one of those classes. To remove an event's dispatch class so that it can run in any class, enter the class as a space enclosed by double quotes. For example:
	`ZA EV 46 NEWCL ""`
	`ZA EV 46 NEWCL B`
	`ZA EV 46 NE (F,V,Q,R,S,T)`

NEWGrp	Updates the event's group ID.

Parameter	Description
NEWUser	Updates the user ID associated with the event. For example, this command changes the user ID to ABC123 for all events that have an application ID of ASG1: `ZA APP ASG1 NEWU ABC123`

NEWSCHENV	Changes the selected event's scheduling environment requirement.
	Examples:
	This command changes the scheduling environment to PROD_DB_AVAIL for event 33:
	`ZA EV 33 NEWSCHENV=PROD_DB_AVAIL`
	This command removes the scheduling environment requirement for event 33:
	`ZA EV 33 NEWSCHENV=`

NEWSysid	Updates the system ID where the job event is executed to the specified ID. Or, you can specify *ANY (which Zeke interprets as a *pool* of all Zekes in the Zekeplex and enables the event to be dispatched on any of the Zeke systems). For example:
	`ZA EV 45 SYS B NEWSYS A`
	`ZA EV 45 SYS B NEWSYS *ANY`

NEWTARget	Updates the Netregid of the system where the job event is executed. For example, this command changes the target information for all job events with jobnames beginning with CER and target of AANAS400 to RMTSYS2: `ZA JOB *CER TARG=AANAS400 NEWTARG=RMTSYS2` **Note:** You cannot alter the target for a job that has been downloaded to Zeke Agent.
NORERun	Removes the RERUN designation. See page 292 parameter for more information on the RERUN parameter.
NOTAFter	Updates the event's 'not after' time (in *hhmm* format). The valid values range from 0000 through 4759. For example: `ZA EV 1 NOTAF 1600`
NOTDUROk	Removes the NOTDURING requirements from an event. For example: `ZA EV 77 NOTDUROK`

Parameter	Description
NOTRig	Updates an event so that it does not trigger other events. For example: `ZA JOB JOBC NOTRIG` **Note:** The NOTRIG option overrides the TRIGOPT option (see page 295). WEAK conditions still can be satisfied by an event that has been ZALTER'd using the NOTRIG parameter. For example, normally, if JOBC is in the schedule, the WHEN condition WHEN (WEOE JOBC) would not be weakly satisfied and would not be marked with a #. However, if JOBC is ZALTER'd with the NOTRIG option, the WHEN condition is weakly satisfied and marked with # because the NOTRIG parameter causes JOBC to be completely ignored for triggering purposes. Since Zeke *does not find* JOBC when checking weak conditions, all weak conditions that refer to JOBC are weakly satisfied. **Note:** For permanent events, you cannot issue this parameter; Zeke assumes that permanent events always participate in triggering.
PRiority	Updates the job OPSYS priority. The valid values range from `1` through `9`. For example: `ZA JOB *AR PRI 4`
RERUN	Adds the RERUN designation to the SQR. The RERUN designation appears in the ZDISPLAY output and is passed to the user exit ZEKE14D. If the Zeke generation option TrigRrn is set to `N`, the event will not trigger the WHEN conditions of other events. You use the NORERUN parameter of the ZALTER command to remove the RERUN designation.
RUN	Satisfies these conditions for an event: • TIMEOK • WHENOK • OPEROK • NOTDUROK • ZRELEASE

Parameter	Description
	Checks these conditions before dispatching the event: • SYSTEM HOLD • INITIATOR • TAPES • RESOURCES. **Note:** NOTDURING clauses are ignored. For example: `ZA JOB TS01H001 RUN`
SChed	Updates the time an event is scheduled to run (SCHEDULE time). The valid values range from `0000` through `4759`. For example: `ZA EV 432 SC 1200`
SENDCOMM	(For use in an XCFONLY recovery.) Forces a COMM record for the specified event(s) to be sent to all other systems in a Zekeplex, even if no changes have been made to the event(s). For example: `ZA EV ABC* SENDCOMM` If a Zekeplex is running in XCFONLY mode, and if a Zeke on one of the system fails, automated recovery is not possible. Depending on the type of failure, scheduling data may become stale. If you see that events are not scheduling properly, you can use the SENDCOMM option of the ZALTER command to manually broadcast SQR refresh requests across the Zekeplex. You can also use the ZPLEX RELOAD COMM command to force a schedule reload and in-memory tables refresh.
Success	Forces an event to F/S (Forced Success) status. F/S satisfies the WHEN conditions for any events that reference that event. For example: `ZA JOB TS01H001 Succ`
SYNch	Synchronizes the SQRs downloaded to a Zeke Agent when the job status on Zeke does not match the job status on Zeke Agent. `LOCAL` Use this keyword when the Zeke copy of the SQR is out-of-date, as indicated by STALEZ in the DL Status field in Schedule View. `ZALTER JOBA SYNCH=LOCAL` Zeke sends a request to Zeke Agent for the updated information and rebuilds the SQR.

Parameter	Description
	`REMOTE` Use this keyword when the Zeke Agent copy of the SQR is out-of-date, as indicated by STALEA in the DL Status field in Schedule View. `ZALTER JOBA SYNCH=REMOTE` Zeke downloads the updated SQR to Zeke Agent.
`TApes`	Change the number of available tape drives required before an event can be dispatched. If you enter zero, the system does not check for tape drives. For example: `ZA EV 62 TA 2`
`TIMEOk`	Satisfies the time requirement (regardless of the actual time). For example: `ZA JOB *AR TIMEOK`
`TImes`	Updates the number of times Zeke dispatches the event per schedule run. The valid values range from `1` through `9999`. For example: `ZA EV 62 TI 4` **Note:** Increasing the count causes the event to run the indicated number of times in this schedule run, but does not make the SQR a recurring event (which must be set up in the EMR). For a recurring event, Zeke resets the WHEN conditions at dispatch time, and the event can become WHEN-satisfied even while it is active. **Note:** For permanent events, do not set a Times value; permanent events can run an unlimited number of times. If a permanent event is later changed to be non-permanent, Zeke automatically sets the Times value to `1`.
`TRIG`	Reverses the effect of the NOTRIG parameter (see page 292) if this parameter is specified before the event is dispatched. The event now can trigger other events. For example: `ZA EV 6 TRIG` **Note:** For permanent events, you cannot issue this parameter; Zeke assumes that permanent events always participate in triggering.

Parameter	Description
TRIGOpt	Applies to recurring events only. A recurring event is one for which the Times field is greater than 1 (i.e., an event that is dispatched more than once per schedule run). This option changes the setting of the Trig field in the SQR. The Trig value indicates when the recurring event can satisfy WHEN conditions (i.e., serve as a trigger) for other events. A non-recurring event is one for which the TIMES value is 1; non-recurring events trigger regardless of the code entered in the Trig field. **Note:** The NOTRIG option (see page 292) overrides the TRIGOPT option. ALL — (Default) The recurring event can trigger other events each time it runs. **Note:** Permanent events (i.e., recurring events which can occur an unlimited number of times) always trigger on *all* occurrences. FIRST — The recurring event can only trigger other events the first time it runs. LAST — The recurring event can only trigger other events the last time it runs. For example, suppose you have a recurring event that is scheduled to run every hour starting at 8:00 and ending at 12:00. If Trig is set to F, the event can satisfy WHEN conditions only on the 8:00 run, and all subsequent trigger calls for this event are ignored (until the event is rebuilt or refreshed). If Trig is set to L, the event can satisfy WHEN conditions only on the 12:00 run, and all other trigger calls for this event are ignored (until the event is rebuilt or refreshed). If Trig is set to A, the event can satisfy WHEN conditions on each run from 8:00 to 12:00.

WHENANd	Adds a specified condition in addition to the existing WHEN conditions.
	
	Examples:
	
	This command adds the specified clause to the existing WHEN condition with an AND relationship (i.e., the added clause must be satisfied in addition to the existing WHEN condition):
	
	`ZA EV 5 WHENAND (EOJ JOB1)`
	
	This command adds a remote WHEN condition with an AND relationship:
	
	`ZALTER EV 1 WHENAND (EOJ JOBC AT SYSB)`

Parameter	Description

WHENOK	Satisfies prerequisites (WHEN conditions) for the specified event. If the event is time satisfied when this parameter is used, the event moves from the schedule queue to the dispatch queue.
	
	Note:
	If there are existing WHEN conditions, you must add any additional WHEN conditions *before* issuing a WHENOK. When you issue a ZALTER WHENOK, Zeke will not recognize subsequently added WHEN conditions.
	
	Examples:
	
	This command satisfies all WHEN conditions for event 1:
	
	`ZA EV 1 WHENOK`
	
	This command satisfies the remote WHEN condition:
	
	`ZALTER EV 65 WHENOK (EOJ JOBB AT SYSB)`
	
	This command satisfies the EOJ JOB1 WHEN condition for event 5. Any other WHEN conditions for event 5 are not affected:
	
	`ZA EV 5 WHENOK EOJ JOB1`
	
	This command satisfies the WHEN condition for the variable $GO for all job events whose jobnames begin with PAY (the value of $GO is not changed):
	
	`ZA JOB *PAY WHENOK VAR $GO`
	
	This command satisfies the EOJ JOBB WHEN condition for event 1 on the remote system SYSB:
	
	`ZALTER EV 1 WHENOK (EOJ JOBB AT SYSB)`
	
	This command satisfies the specified WHEN condition for version 2 of event 5 (parentheses are required when using the VER keyword):
	
	`ZALTER EV 5 VER 2 WHENOK (EOJ JOBA VER 2)`
	
	This command satisfies the specified multiple WHEN conditions for event 25 (any other WHEN conditions for event 25 are not affected):
	
	`ZALTER EV 25 WHENOK (EOJ JOBA, WEOJ JOBB, AEOJ JOBC)`

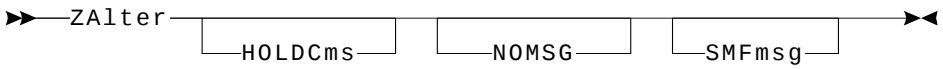
Parameter	Description

WHENOR	Adds a specified WHEN condition to the existing WHEN conditions with an OR condition.
	Examples:
	This command adds the EOJ JOB1 WHEN condition to the existing WHEN conditions for event 5 (the event is WHEN satisfied when either the existing WHEN condition or this newly added one is satisfied):
	`ZA EV 5 WHENOR (EOJ JOB1)`
	This command adds the specified WHEN conditions to the existing WHEN conditions for event 5:
	`ZA EV 5 WHENOR '(EOJ JOB1 and EOJ JOB2) OR (VAR $A EQ GO)'`
	The event is WHEN satisfied when one of these conditions is satisfied:
	• Existing conditions
	• EOJ JOB1 and EOJ JOB2
	• VAR $A EQ GO
	This command adds a remote WHEN condition with an OR relationship.
	`ZALTER EV 1 WHENOR (EOJ JOBC AT SYSB)`

Altering Message Generation

You can temporarily override certain types of Zeke message generation.

Syntax



Parameters

You can use these parameters with the ZALTER command:

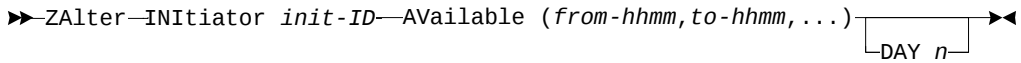
Parameter	Description
HOLDCms	Turns off messages to the console and places a hold on any events that would have generated this message. Otherwise, messages appear on the operator console if Zeke stops processing JCL requests. For example: `ZA HOLDCMS`
NOMSG	Suppresses messages Z0905I and Z0906W.

Parameter	Description
SMFmsg	Toggles on/off the trace messages related to Zeke calls to SMF exits. For example: `ZA SMFM` See "Displaying Tracing Messages" on page 351 for more information on trace options and settings related to SMF exit processing.

Altering Initiator Availability

Zeke maintains a table that contains the times each initiator is available. You specify the days and times each initiator is available to Zeke through the online facility. Use the ZALTER INITIATOR command to temporarily override the table. To reset the times, issue the ZRELOAD INIT command.

Syntax



Parameters

You can use these parameters with the ZALTER command:

Parameter	Description
AVailable	Required. Updates the times an initiator is available. You specify the start and stop times in parentheses, separated by a comma. You can enter multiple start and stop times (up to four ranges). `ZA INI T1 AV (0,400,0800,2400)` Make initiator T1 available to Zeke from 00:00 to 4:00, and from 8:00 to 24:00.
DAY	Updates the day the initiator is available. The valid values range from `1` (Monday) through `7` (Sunday). If you do not include this parameter, the current day is assumed. `ZA INI T1 AV (0,2400) DAY 3` Make initiator T1 available to Zeke from 00:00 to 24:00 on Wednesday.

Parameter	Description
INItiator	Required. Updates the named initiator ID. `ZA INI T1 AV (0,2400)` Make initiator T1 available to Zeke from 00:00 to 24:00. **Note:** If you are using JES3 or generation option DispSel is set to `N`, the ZALTER INITIATOR command is not valid.

ZDELETE Command

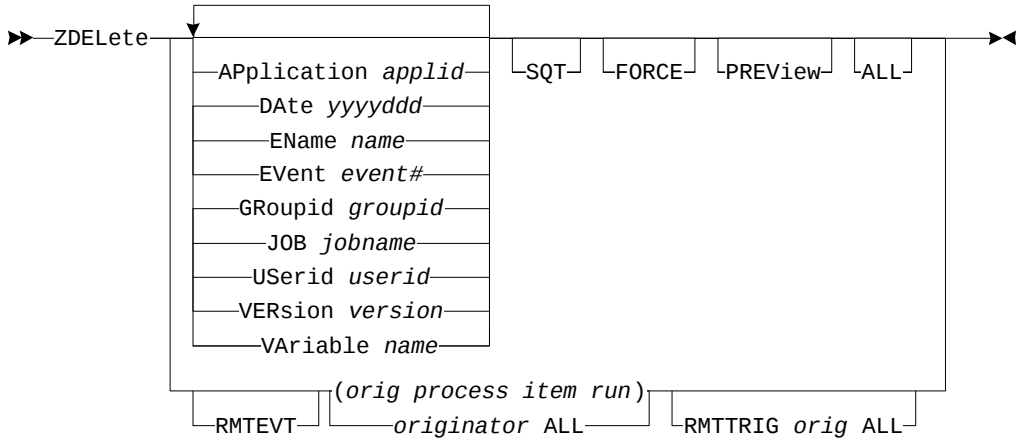
You use the ZDELETE command to delete these types of elements (depending on the parameters you use):

- Variables in the Zeke database. Consider these points when deleting variables:
  - Zeke issues a console message to confirm successful deletion of the selected variables.
  - Because Zeke variables are added to the Zeke database when they first are used, you can re-add deleted variables by including them in a ZEKESET SET VAR statement or in the ZSET operator command.
- Scheduled events (SQRs). Consider these points when deleting scheduled events:
  - Zeke issues a console message to confirm successful deletion of the selected SQRs.
  - Generally, when you specify criteria for selecting the SQRs that you want to delete, Zeke searches (in schedule time sequence) the scheduled events that have not been dispatched and deletes the first matching SQR.
  - To ensure that you delete only the desired SQRs, ASG recommends that you use the full SQR key (i.e., event number, schedule date, and version number), or locate the event in Schedule View and issue the DEL line command to delete it.
  - You can use the PREVIEW parameter to help prevent the unintentional deletion of a large number of events.
  - You might want to consider allowing operators to use only the ZDISABLE command for this type of control over scheduled events (and restricting the use of the ZDELETE command to system administrators only).
  - After an SQR has been deleted, you can re-add a new SQR for the same event using the ZADD command. The new SQR reflects any information that has been updated in the EMR.
- You can delete permanent events (which run an unlimited number of times) only with the ZDELETE command.Remote events in the Zekeplex. Consider these points when deleting remote events:
  - Delete remote events only if the events cannot be completed and then deleted from the Zekeplex automatically. If a remote event is deleted and its corresponding task still exists on the originating system, the originator task will require manual adjustment since the originating system no longer will receive any status updates after the remote event is deleted.

- You use the ZDISPLAY RMTEVT command to display the identifying information for the remote event you want to delete. Because the ID values are displayed in the same order as you specify them on the ZDELETE RMTEVT command, you can copy the values from the ZDISPLAY output and paste them inside parentheses for the ZDELETE RMTEVT command.

- Remote triggers in the Zekeplex from a specific originating system. Consider these points when deleting remote triggers:
  - Because remote triggers for an originating system are automatically updated when the system reconnects with the Zekeplex, delete all remote triggers only if the system is not expected to be reconnected.
  - If remote triggers are deleted in error while the originating system is connected to the Zekeplex, the triggers are restored when the system disconnects and reconnects. This can be forced by disabling and re-enabling the originator agents (e.g., Zena agents) for the Zekeplex.

Syntax



Parameters

You can use these parameters with the ZDELETE command:

Parameter	Description

ALL	When you use the ALL parameter, Zeke searches all scheduled events (regardless of their status), and deletes all that match the selection criteria. For remote events or triggers, Zeke deletes all remote events or triggers in the Zekeplex.
	Examples:
	`ZDELETE ALL`
	`ZDEL RMTEVT F79B53A264A8 ALL`
	Without the ALL parameter, Zeke searches scheduled events that have not been dispatched (in schedule time sequence) and deletes only the first event that matches the selection criteria.

APplication	Deletes events based on the specified application ID.
	Note:
	You can use wildcard and placeholder characters in your selection criteria. See "Wildcard and Placeholder Characters" on page 259.
	Examples:
	This command deletes the first event in the schedule that matches an application ID of PAY:
	`ZDEL APPL PAY`
	This command deletes the first event in the schedule that matches an application ID beginning with P in position 1, any letter in position 2, and Y in position 3:
	`ZDEL APPL P?Y`
	This command deletes the first event in the schedule that matches an application ID beginning with AY:
	`ZDEL APPL AY*`

DAte	Deletes events based on the specified date (in *yyyyddd* or *yyddd* format) (in the case of multiple occurrences of the event on different dates). For example: `ZDEL EV 10 DATE 2020001` When using *yyddd* format, if *yy* is greater than or equal to `50`, it is considered a twentieth century date (`19`*xx*); if it is less than `50`, it is considered a twenty-first century date (`20`*xx*).

Parameter	Description
DOne	Searches (in schedule time sequence) events that have not been dispatched (i.e., Pending, Active, Done) or disabled, and deletes the first SQR that matches the selection criteria. If you include the ALL parameter, Zeke searches events that have not been dispatched and deletes all SQRs that match the selection criteria.

ENAMe	Deletes events based on the specified event name.
	
	Note:
	You can use wildcard and placeholder characters in your selection criteria. See "Wildcard and Placeholder Characters" on page 259.
	
	Examples:
	
	This command deletes the first event in the schedule that matches the event name EVNTNAM1:
	
	`ZDEL ENAME EVNTNAM1`
	
	This command deletes the first event in the schedule that matches an event name beginning with the character string EVNTNAM and having any letter in position 8:
	
	`ZDEL ENAME EVNTNAM*`

EVent	Deletes all SQRs (regardless of their statuses) for the event with the specified event number.
	
	Examples:
	
	`ZDELETE EV 5`
	
	`ZDEL EV (3,5)`

FORCE	Releases acquired resources for the events to be deleted (before deleting the events). FORCE does not release kept resources. For example: `ZDEL EV 12 FORCE` Using this parameter produces the same result as releasing all of the resources and then deleting the event. If there are no resources for the event, this parameter is ignored.

Parameter	Description

GRoupid	Deletes events based on the specified group ID.
	Note:
	You can use wildcard and placeholder characters in your selection criteria. See "Wildcard and Placeholder Characters" on page 259.
	Examples:
	This command deletes the first event in the schedule that matches the group ID OPR:
	`ZDEL GROUP OPR`
	This command deletes the first event in the schedule that matches the group ID beginning with O in position 1, any letter in position 2, and R in position 3:
	`ZDEL GROUP O?R`
	This command deletes the first event in the schedule that matches the group ID beginning with OP:
	`ZDEL GROUP OP*`

JOB	Deletes events based on the specified jobname.
	Note:
	You can use wildcard and placeholder characters in your selection criteria. See "Wildcard and Placeholder Characters" on page 259.
	Examples:
	This command deletes events the first event in the schedule that matches the jobname TESTJOB1:
	`ZDEL JOB TESTJOB1`
	This command deletes the first event in the schedule that matches a jobname beginning with the character string TESTJOB and any letter in position 8:
	`ZDEL JOB TESTJOB?`
	This command deletes events in the schedule that have a jobname ending in B1:
	`ZDEL JOB *B1`

PREView	Produces a list of events that would be deleted from the schedule (based on the command string) without executing the command. This option helps prevent unintentional deletion of events. For example, this command displays a preview of events to be deleted from the schedule that match based on the parameters GROUP OPR: `ZDEL GROUP OPR PREVIEW`

Parameter	Description

RMTEVT	Deletes remote events from the Zekeplex. You can delete a specific remote event or all remote events from a specific originating system.
	Note:
	Delete remote events only if the events cannot be completed and then deleted from the Zekeplex automatically. If a remote event is deleted and its corresponding task still exists on the originating system, the originator task will require manual adjustment since the originating system no longer will receive any status updates after the remote event is deleted.
	To delete a *single* remote event, you must specify the originator ID, process ID, item ID and run ID.
	`ZDEL RMTEVT (originator process item run)`
	where:
	originator is the system that submitted the event to the Zekeplex.
	process is the process ID of the task on the originating system.
	item is the item ID of the task on the originating system.
	run is run ID of the task on the originating system.
	Example:
	`ZDEL RMTEVT (F79B53A264A8 0000000407 0000000001 BAA8805034E9)`
	Note:
	You use the ZDISPLAY RMTEVT command to display the identifying information for the remote event you want to delete. Because the ID values are displayed in the same order as you specify them on the ZDELETE RMTEVT command, you can copy the values from the ZDISPLAY output and paste them inside parentheses for the ZDELETE RMTEVT command.
	To delete *all* remote events submitted by a specific originating system, you must specify the originator ID:
	`ZDEL RMTEVT originator ALL`
	Example:
	`ZDEL RMTEVT F79B53A264A8 ALL`

RMTTRIG	Delete all remote triggers from the Zekeplex that were sent by the specified originating system. You must specify the originator ID:
	`ZDEL RMTTRIG originator ALL`
	Example:
	`ZDEL RMTTRIG F79B53A264A8 ALL`

Parameter	Description
SQT	Deletes the matching schedule queue table (SQT) entries on the system where the ZDELETE command is issued. Schedule table entries on other Zeke systems that share the database are not deleted. SQRs in the Zeke database are not deleted. Use this option in these situations: • After an event has been deleted forcibly using the EVENT DELETE FORCE command. (You must issue the ZDELETE SQT command on all Zeke systems that share the database.) • If the schedule tables no longer are synchronized with the SQRs in the database. (This condition is considered an error; contact ASG Customer Support for assistance.) For example, if the ZDISPLAY output displays a particular schedule record, but the ZDISPLAY SQR output does not display the same record, then you can use the ZDELETE SQT parameter to remove the record from the schedule tables. **Note:** If this parameter is used unintentionally, you can issue the ZRELOAD SCHD command to re-add the record to the schedule tables.

USerid	Deletes events based on the specified user ID.
	
	Note:
	You can use wildcard and placeholder characters in your selection criteria. See "Wildcard and Placeholder Characters" on page 259.
	
	Examples:
	
	This command deletes the first event in the schedule that matches the user ID of DEV:
	
	`ZDEL USER DEV`
	
	This command deletes the first event in the schedule that matches a user ID beginning with D in position 1, any letter in position 2, and V in position 3:
	
	`ZDEL USER D?V`
	
	This command deletes the first event in the schedule that matches a user ID beginning with EV:
	
	`ZDEL USER EV*`

VAriable	Deletes the specified variable. For example:
	
	`ZDELETE VAR $XZY`
	
	`ZDEL VAR $MSGON`

Parameter	Description

VERsion	Deletes the specified event version. If no version number is specified, then all versions of the event are deleted.
	Examples:
	This command deletes all SQRs in the schedule for version 2 of event 30.
	`ZDEL EV 30 VER 2`
	This command deletes only the SQR for schedule date January 1, 2020 for version 2 of event 30.
	`ZDEL EV 30 DATE 2020001 VER 2`

ZDISABLE Command

The ZDISABLE command enables you to disable events, automatic replies, or electronic vaulting.

Disabling Scheduled Events

Disabling an SQR does not affect future scheduling of that event. As with all operator commands, the EMR is not changed.

A display of done events includes disabled events, which are noted on the screen.

Disabled events are deleted from the schedule, and dropped at the same time other completed events are dropped (typically when the next day's schedule update is processed).

WHEN conditions are not satisfied for a disabled event.

An active job that is disabled using the ZDISABLE command continues to run to completion, but Zeke ignores it for the purposes of triggering and no longer tracks it. Such a job appears in Schedule View with an ACTIVE DISABLED status, even after it is done.

To enable a disabled event, use the ZENABLE command.

Syntax



Parameters

Use any of the global event selection parameters (starting on page 261) with the ZDISABLE command.

Examples

These are examples of using the ZDISABLE command and some of the global event selection parameters.

This command disables the SQR for event 62:

```
ZDISABLE EV 62
```

This command disables the SQR for event 62 that is dated January 1, 2020:

```
ZDISA EV 62 DA 2020001
```

This command disables all job events with a jobname beginning with PR for schedule date January 1, 2017 (these are the prior day's jobs that are being removed from the schedule manually):

```
ZDISA JOB *PR DA 2020001
```

Note:

The ZDISABLE command prevents WHEN conditions referring to that event from being satisfied.

Disabling Automatic Replies

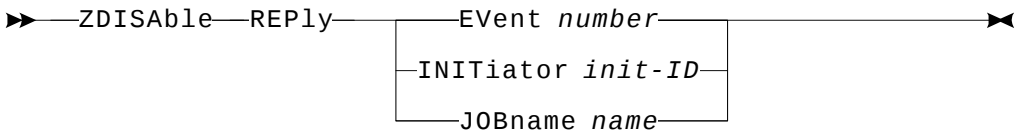
You can use the ZDISABLE command to disable the auto reply capability of an event or initiator/partition.

To disable an auto reply for an event that is not running, use the REPLY and EVENT parameters.

Note:

See the *ASG-Zeke Scheduling for z/OS User's Guide* for more information on disabling auto replies.

Syntax



Parameters

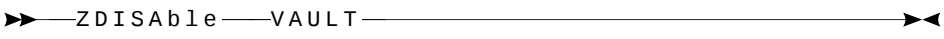
You can use these parameters with the ZDISABLE command to disable automatic replies:

Parameter	Description
EVent	Specifies the event number for which the auto reply is to be disabled. The auto reply can only be disabled before the event is dispatched. If the event is running, use the INIT keyword. For example, this command disables auto reply for event 77: `ZDISA REP EV 77`
INITiator	If the event is running, you must disable the auto replies for that initiator/partition. Enter the initiator ID with the REPLY and INIT parameters. For example, this command disables the auto reply currently active in initiator T3: `ZDISA REP INIT T3`
JOBname	Jobname for which you want to disable automatic replies.
REPly	Signifies that you are using ZDISABLE to disable replies (as opposed to disabling events or vaulting).

Disabling Electronic Vaulting

You can disable electronic vaulting while Zeke is active using the ZDISABLE command. (See "VAULT—Control the Vault Dataset" on page 113 for syntax for performing the function in batch.)

Syntax



Parameters

Use this parameter with the ZDISABLE command to disable electronic vaulting:

Parameter	Description
VAULT	Disables electronic vaulting.

ZDISPLAY Command

The ZDISPLAY command displays these types of requested information (which are described in more detail in the following sections):

- Information about events (e.g., statuses, times, etc.).
- Value and current status of a Zeke variable.
- Values of event variables.
- Times that initiators are considered available, along with status information.
- Automatic replies that are active for a job event.
- System IDs contained in a system pool.
- CPUs that are currently running.
- Tracing calls to the schedule and variable monitors and to Zeke's security routine.
- Event predecessor and event successor information for an event that has other events dependent on it (or for an event that is dependent on others).
- Catalog and vault volume and dataset information.
- Information about remote Zeke and Zena agents, remote events and triggers, remote variables, or download agents and events that have been downloaded.

ZDISPLAY Formats

Zeke provides several display formats. The format is determined by the parameters you specify with the ZDISPLAY command.

Each display format contains a message number and column headings (which are defined with each display format sample in the following sections).

Displaying Event Information

Event-related parameters enable you to obtain various information about event definitions, scheduling, and dispatching (e.g., event status, schedule times, JCL sources, etc.). Some event parameters are used to select a set of events with common attributes or statues and display that information, while others are used for displaying information about a particular event.

The ZDISPLAY command also is useful for previewing the results of another command (especially if the command includes any wildcards). For example, suppose you are planning to disable a group of events using this command:

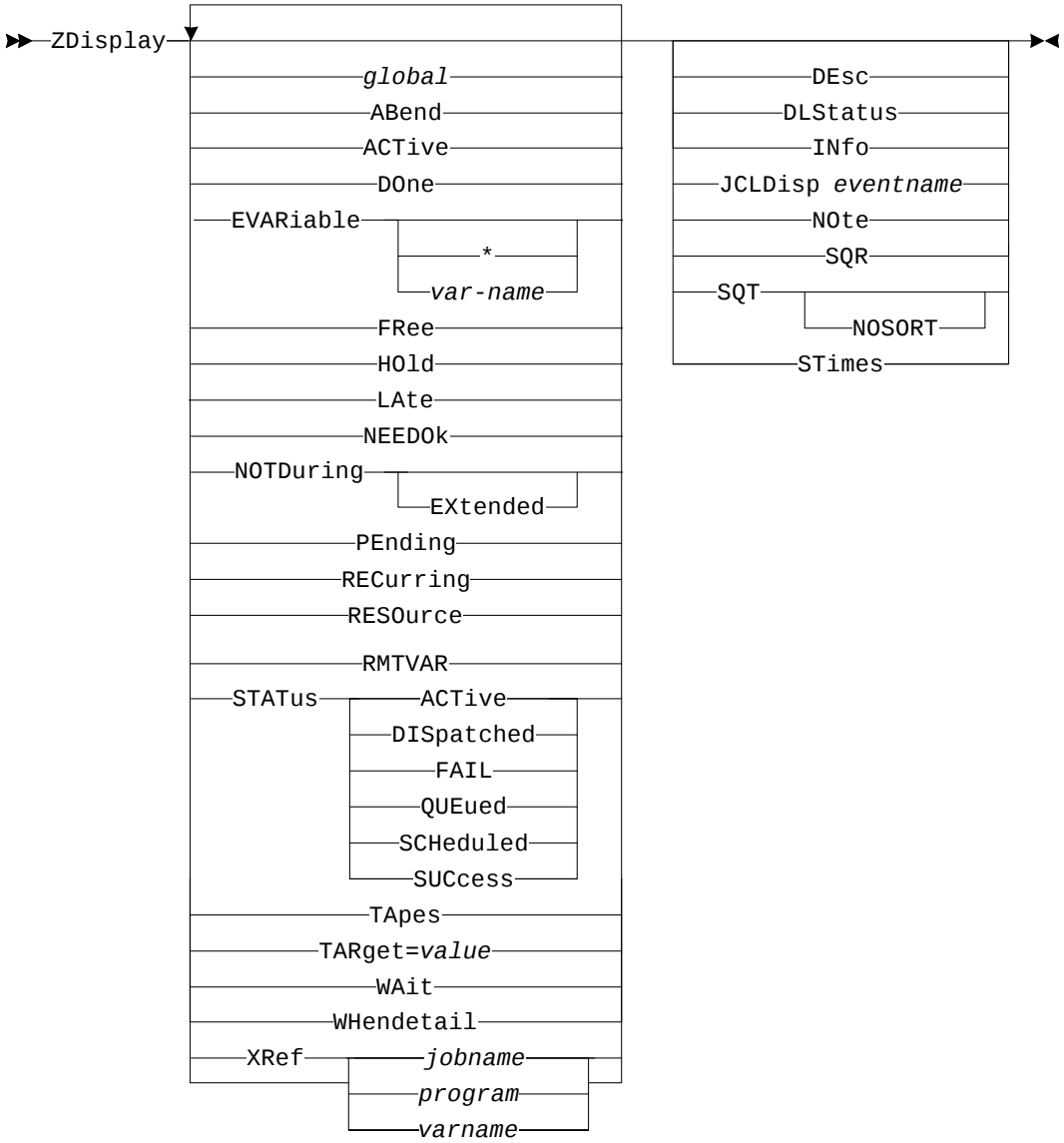
```
ZDISABLE JOB *PR
```

First, you could issue this command to verify that the intended events are affected:

```
ZDISPLAY JOB *PR
```

Syntax

This diagram shows the command format for issuing ZDISPLAY with event-related parameters:



Parameters

To request event information, you can use both the event-related parameters as described in this table and the global event selection parameters (starting on page 261):

Parameter	Description
ABend	Selects the events that were dispatched and terminated abnormally.
ACTive	Selects all events that are currently processing.
DEsc	Displays the description for the selected event.
DLStatus	Displays the event download status.

DOne	Selects only scheduled events that are completed.
	Examples:
	This command displays all completed events:
	`ZD DO`
	This command displays completed job events:
	`ZD JOB DONE`

EVARiable	Selects scheduled events that contain an event variable(s). These are the options parameters:
	`*` Selects all scheduled events that contain an event variable(s).
	`name` Specifies the name of the event variable. The preceding dollar sign ($) is optional.
	Examples:
	Either of these commands display all scheduled events that contain event variables:
	`ZD EVAR`
	`ZD EVAR *`
	Either of these commands display the schedule events that contain the event variable named $MYVAR:
	`ZD MYVAR`
	`ZD $MYVAR`

FRee	Selects only events that are not on operator hold (this includes completed events). For example, this command selects job events that are not on operator hold (this includes completed job events): `ZD JO FRE`

Parameter	Description

HOld	Selects only events that are on hold. For example, this command selects job events on hold:
	`ZD JOB HO`
	The HOLD parameter can be used with a qualifier such as OPER, SJCL, NET, RFSH, DNPL, PSID, INTR or REXX to select events that are on hold for a particular reason.
	For example, this command selects events on hold with a job card or JCL submittal problem:
	`ZD HOLD=SJCL`
	This command selects REXX events on hold with a REXX dispatch error:
	`ZD HO=REXX`

INfo	Displays the application, group, and user ID instead of the run date, dispatch priority, frequency, and count. Display Format 4 is used (see page 325).
JCLDisp	Displays the JCL source for a specified event. For example, this command displays the JCL source for event 2: `ZD JCLD EV 2`

LAte	Selects late events. An event is late if it was not dispatched before its specified late start time (i.e. Latestart value) or if it did not complete by its 'late end' time (i.e., Lateend value).
	For example, this command displays all late events:
	`ZD LATE`
	This command displays late job events:
	`ZD JOB LA`
	This command displays late job events in the dispatch queue:
	`ZD JO LA DQ`

NEEDOk	Selects only events that require an operator OK. For example, this command selects Zeke command events that require an operator OK: `ZD ZC NEEDOK`
NOSORT	Displays the selected events in the order they appear in the schedule table instead of sorting them by schedule time. **Note:** Using this option can help speed up ZDISPLAY processing for a large number of events.

Parameter	Description
NOTDuring	Selects events with one or more NOTDURING conditions specified.
NOte	Displays up to six lines of information about the event (if notes exist in the event documentation).

PEnding	Selects jobs that are pending.
	Examples:
	This command displays all pending events:
	`ZD PE`
	This command displays pending job events:
	`ZD JOB PE`

RECurring	Selects only recurring events (i.e., events that are dispatched more than once per schedule run). For example, this command selects recurring job events: `ZD JOB REC`
RESOurce	Selects the events that have resource requirements. For example, this command displays Zeke command events that have resource requirements: `ZD ZC RESO`
SQR	Displays diagnostic information about the schedule records in the Zeke database for the specified event numbers. **Note:** The ZD SQR and ZD SQT commands enable you to determine whether the schedule table entries match the schedule records in the Zeke database. The output includes this information for the selected events: • Event number, date (in Julian format), type, and jobname (if any) • Early, schedule, and late times • Dispatch frequency (if any) and number of times to be dispatched • Whether the time and WHEN conditions are satisfied • Whether the event is in the dispatch queue (or already completed) • Whether the event has passed its late time • Whether the event is on hold

Parameter	Description
SQT	Displays diagnostic information about matching schedule queue table (SQT) entries. **Note:** The ZD SQT and ZD SQR commands enable you to determine whether the schedule table entries match the schedule records in the Zeke database. The output includes this information: • Event number and version number • Schedule date • Whether the event was scheduled manually • Address of scheduling control block • Sequencing control codes
STATus	Selects the events that have the specified status. These are the valid statues: `ACTive` Displays events currently running. `DISpatched` Displays events that have been dispatched, but have not started running. `FAIl` Displays events that ended abnormally. `QUEued` Displays events that are in the dispatch queue. `SCHeduled` Displays events with time and WHEN conditions that have not been satisfied. `SUCcess` Displays events that completed successfully. For example, this command displays all events that completed successfully: `ZD STAT SUC`
STimes	Displays the values for each of the event's defined schedule times (e.g., early, schedule, 'late start', 'must end', 'not after', average duration, etc.).

TApes	Selects job events that use tape drives. The JOB parameter is assumed.
	Examples:
	This command selects all job events that use tape drives:
	`ZD TAPES`
	This command selects all job events that use tapes and are late:
	`ZD TAPE,LATE`

Parameter	Description
TARget	Selects job events with the specified target. For example, this command displays all job events with a Target value of *REMOTE specified in its EMR: `ZD TARGET=*REMOTE`

WAit	Selects events waiting in the dispatch queue. The command output uses Display Format 3 (see page 329)
	The reason each event is waiting in the dispatch queue is displayed next to each event. See "Event Status/Reason Code" on page 443 for descriptions of the reason codes.
	This optional keyword also can be included:
	EXtended — For any selected event that is waiting because of a NOTDURING condition, this keyword displays detailed information about the job or program that currently is preventing the event from being dispatched.
	Examples:
	`ZD WAIT`
	Selects all events in DQ status, and displays the WAIT reasons.
	`ZD WA,JOB`
	Selects all job events in the dispatch queue and displays the reason why each event is waiting.
	`ZD WAIT EX`
	Selects events waiting in the dispatch queue and, for any event with a NOTDURING condition, identifies the job or program that is causing the wait.

WHendetail	Selects events with WHEN condition statements and displays the status of the prerequisite. (Additional lines are displayed for additional prerequisites.) Use this parameter to determine which conditions are delaying the dispatching of an event. For example: `ZD EV 13 WH` These are the possible status codes that are displayed for each condition: * — Zeke satisfied the WHEN condition. + — An operator command was used to manually satisfy the WHEN condition. # — A weak WHEN condition was satisfied because the event to satisfy the condition is not in the schedule.

Parameter	Description

XRef	Selects events with WHEN conditions that contain the specified jobname, program name, or variable. This produces the same display format as the WHENDETAIL parameter.
	Examples:
	This command displays WHEN information for events triggered by the variable $VAR1:
	`ZD XREF $VAR1`
	This command displays WHEN information for events triggered by job JOBA:
	`ZD XR JOBA`
	This command displays WHEN information for events triggered by program PROGA:
	`ZD XR PROGA`

Main Event Display Format

Generally, Zeke displays command output in its main format when you issue the ZDISPLAY command with event parameters.

Zeke displays events in sequence starting with the earliest schedule time. An asterisk (*) to the right of the event number indicates that the schedule entry is in the dispatch queue, ready for dispatching. An asterisk (*) to the left of a jobname, program, or variable indicates that the WHEN condition is satisfied.

These symbols are displayed to the left of the STATUS column on each screen:

Symbol	Meaning
blank	The event is neither WHEN nor time-satisfied.
*	The event is WHEN and time-satisfied and ready for dispatching.
$	The event is in the dispatch queue and is waiting for a free initiator.
-	The event has no WHEN condition, but is not time-satisfied.
W	The event's WHEN conditions are satisfied, but the event is waiting to be time-satisfied.
T	The event is time-satisfied, but is waiting for WHEN conditions to be satisfied. Use the ZDISPLAY WHEN command to display the WHEN conditions of the event.

Symbol	Meaning
#	This symbol appears to the left of a weak WHEN condition. This indicates the WHEN condition is satisfied because the dependency is not in the schedule.

This is an example of the main display format:

```
Z0922I DATE    RDATE   VER TYPE JOB/EVT NAME DP SCHED FREQ  CNT    STATUS
000003 2020325 2020325 001 JOB  NEWTST03     50 00:00          1 * SUCC 15:46
000003 2020325 2020325 003 JOB  NEWTST03     50 00:00          1 * SUCC 14:06
000003 2020325 2020325 004 JOB  NEWTST03     50 00:00          1 * SUCC 14:09
000010 2020329 2020329 001 JOB  CER6K1       50 00:00          1 * SUCC 16:12
000014 2020329 2020329 001 JOB  CERJ14       50 00:00          1 * HOLD
000003 2020328 2020328 001 JOB  NEWTST03     50 00:00 00:01  123 T
000010 2020329 2020329 002 JOB  CER6K1       50 00:00          1 * PEND 16:17
000003 2020332 2020332 001 JOB  NEWTST03     50 00:00          1 T
000004*2020332 2020332 001 JOB  NEWTST04     50 00:00          1 *
```

These are column headings:

Heading	Description
Z0922I	Message ID for the column heading. This column displays the event numbers. An asterisk (*) next to the event number indicates that the event is in the dispatch queue awaiting dispatch (unless it is already completed). An event becomes eligible for dispatch because its time and WHEN conditions both are satisfied.
DATE	Julian date the event was added to the schedule.
RDATE	Run date.
VER	Event version.
TYPE	Event type.
JOB/EVT NAME	Jobname (for a job event); otherwise, the event name.
DP	Dispatch priority code.
SCHED	Normal dispatch time.
FREQ	For recurring events, the amount of time between dispatches.
CNT	Number of times the event is scheduled.

Heading	Description
STATUS	The processing status of the event. See "Event Status/Reason Code" on page 443 for an explanation of each status code. The column to the left of the STATUS column displays one of the symbols in "ZDISPLAY Formats" on page 311. If the job is complete, the completion time is also displayed, to the right of the STATUS column.
{unlabeled}	The last column indicates whether the job was executed at a remote location and lists the name of the remote location as it appears in the Target field on the EMR screen.

Other Event Display Formats

Zeke uses additional display formats, which vary from the main format depending on which parameter you use with the ZDISPLAY command:

- See "Event Display Format—DESC" on page 321
- See "Event Display Format—DLSTATUS" on page 321
- See "Event Display Format—DONE" on page 322
- See "Event Display Format—EVAR" on page 323
- See "Event Display Format—HOLD" on page 324
- See "Event Display Format—INFO" on page 325
- See "Event Display Format—JOB" on page 326 and "Event Display Format—JOB and DONE" on page 327
- See "Event Display Format—NOTE" on page 329
- See"Event Display Format—SUCCESSOR/PREDECESSOR" on page 334
- See "Event Display Format—WAIT" on page 329
- See "Event Display Format—WHENDETAIL" on page 332
- See "Event Display Format—XREF" on page 332

Event Display Format—DESC

This format is used if the DESC parameter is specified. The screen format is the same as the main display format except there is an additional line of information that provides the description for each event.

```
Z0922I  DATE    RDATE    VER TYPE JOB/EVT NAME DP SCHED FREQ  CNT    STATUS
 000003 2020325 2020325  001 JOB  NEWTST03     50 00:00          1 * SUCC 15:46
   DESC: TEST EVENT
 000003 2020325 2020325  003 JOB  NEWTST03     50 00:00          1 * SUCC 14:06
   DESC: TEST EVENT 2
 000003 2020325 2020325  004 JOB  NEWTST03     50 00:00          1 * SUCC 14:09
   DESC: TEST EVENT 3
 000010 2020329 2020329  001 JOB  CER6K1       50 00:00          1 * SUCC 16:12
   DESC: REMOTE JOB - UNIX
 000014 2020329 2020329  001 JOB  CERJ14       50 00:00          1 * HOLD
   DESC: REMOTE JOB - NT
 000003 2020328 2020328  001 JOB  NEWTST03     50 00:00 00:01  123 T
   DESC: TEST EVENT 4
 000010 2020329 2020329  002 JOB  CER6K1       50 00:00          1 * PEND 16:17
   DESC: ROUTED TO UNIX BOX 6K1
```

Event Display Format—DLSTATUS

This format is displayed when the DLSTATUS parameter is specified:

```
Z09D5I DATE    VERSN TYPE EVENT NAME   TARGET   STATUS
000001 2020212 00000 JOB  TESTJOB1     NTAGENT  NOTSNT
  DESC: JOB1
000002 2020212 00000 JOB  TESTJOB2     NTAGENT  NOTSNT
  DESC: JOB2
000003 2020212 00000 JOB  TESTJOB3     NTAGENT  NOTSNT
  DESC: JOB3
Z0905I NUMBER OF SCHEDULE ENTRIES SELECTED WAS 00003 SYSTEM ASGX320A
```

These are the column headings:

Heading	Description
Z09D5I	Message ID for the column heading.
DATE	Julian date of the SQR.
VERSN	Version number of the job event.
TYPE	Event type. Only job events can be downloaded to Zeke Agent, so JOB should always be displayed as the type.
EVENT NAME	Event name.
TARGET	Netregid of the remote system downloading the job event.

Heading	Description	
STATUS	Download status of the job event. These are the valid statuses:	
	NOTSNT	Event has not yet been downloaded to Zeke Agent.
	NOSUPP	The event was not downloaded because it has an attribute not supported for downloaded schedules, such as logical resources or a condition code record.
	ERROR	An internal error occurred during schedule download to target.
	SENT	Job event has been successfully downloaded to Zeke Agent.
	PENDING	An update to the downloaded SQR is pending.
	STALEZ	Event status is mismatched between Zeke and Zeke Agent. The Zeke status is out-of-date.
	STALEA	Event status is mismatched between Zeke and Zeke Agent. The Zeke Agent status is out-of-date.
	??*xx*??	Unknown status. Contact ASG Customer Support for assistance.

Event Display Format—DONE

This format is used if the DONE parameter is specified:

```
Z0969I DATE    VER TYPE JOB/EVT NAME CCODE SCHED FREQ  CNT      STATUS   R
000011 2020234 000 JOB  NEWJOBA      C0000 00:00          1 * SUCC 16:43
000011 2020234 000 JOB  NEWJOBA      C0000 00:00          1 * SUCC 16:24
000012 2020101 001 JOB  NEWJOBB      C0000 00:00          1 T F/FL 15:38
000014 2020239 001 JOB  NEWJOBD      C0000 00:00          1 T DSBL
000021 2020101 000 JOB  NEW806       S806  00:00          1 * FAIL 18:55
000021 2020105 000 JOB  NEW806       S806  00:00          1 * FAIL 10:59
            RERUN
000031 2020093 000 JOB  NEWXDCB      C0000 00:00          1 * FLOK 19:05
000031 2020151 000 JOB  NEWXDCB      S222  00:00          1 * FAIL 17:56
000031 2020123 000 JOB  NEWXDCB      C0000 00:00          1 * SUCC 11:53
000041 2020196 000 JOB  NEWBPXBA     C0000 00:00          1 * SUCC 13:24
000011 2020101 000 JOB  NEWJOBA      C0000 00:01          1 *
000034 2020239 000 JOB  NEWJOBR1     C0000 00:01          1 *
000012 2020171 001 JOB  NEWJOBB      C0000 23:00          1 * SUCC 17:05
000017 2020123 000 REXX REXXEVENT1           00:00          1 * SUCC 15:51
Z0905I NUMBER OF SCHEDULE ENTRIES SELECTED WAS 00014 SYSTEM NEWA
```

These are the column headings:

Heading	Description
Z0969I	Message ID for the column heading. This column displays the event numbers.
DATE	Julian date of the SQR.
VER	Version of the event.
TYPE	Event type.
JOB/EVT NAME	Jobname (for a job event); otherwise, the event name.
CCODE	Abend or condition code.
SCHED	Normal dispatch time.
FREQ	For recurring events, the amount of time between dispatches.
CNT	Number of times the event is scheduled.
STATUS	The completion status and completion time of the event. See "Event Status/Reason Code" on page 443 for an explanation of each status code. The column to the left of the STATUS column displays one of the symbols in "ZDISPLAY Formats" on page 311.
R	An N indicates the job was submitted with remote system specified as the target. If blank, the job was submitted to run locally.

Event Display Format—EVAR

This format is used if the EVAR parameter is specified. The screen format is the same as the main format except there are additional lines of information that list the event variables.

```
Z0922I DATE    RDATE   VER  TYPE JOB/EVT NAME DP SCHED FREQ  CNT    STATUS
000034 2020336 2020336 001  JOB  KAMTST03     50 00:00          1 T
                                    -- VARIABLE --
$MYVAR         =VALUE
$EVTVAR        =YES
Z0905I NUMBER OF SCHEDULE ENTRIES SELECTED WAS 00001 SYSTEM SYSA
```

Event Display Format—HOLD

This format is displayed if the HOLD parameter is specified:

```
Z0922I DATE    RDATE   VER TYPE JOB/EVT NAME DP SCHED FREQ  CNT    STATUS
000014 2020329 2020329 001 JOB  CERJ14       50 00:00          1 * HOLD
000013 2020332 2020332 001 JOB  TESTJOB1     50 00:00          1 * HOLD NET
000009 2020332 2020332 001 JOB  CERJ9        50 00:00          1 * HOLD NET
000013 2020332 2020332 002 JOB  TESTJOB1     50 00:00          1 * HOLD NET
```

These are the column headings:

Heading	Description
Z0922I	Message ID for the column heading. This column displays the event numbers.
DATE	Julian date of the SQR.
RDATE	Run date.
VER	Version of the event.
TYPE	Event type.
JOB/EVT NAME	Jobname (for a job event); otherwise, the event name.
DP	Dispatch priority code.
SCHED	Normal dispatch time.
FREQ	For recurring events, the amount of time between dispatches.
CNT	Number of times the event is scheduled.
STATUS	The last column displays the HOLD status and reason codes. These are the valid statuses: DNPL — Zeke attempted to dispatch a pool event on a VSE system when the DispSel generation option is set to `N`. The event was placed on hold. DSNT — There are multiple SQRs in the schedule with the same event number and the same DSN trigger specified. The Dsntrig generation option is set to `NT`, so Zeke did not trigger any of the events, and the events were placed on hold. INTR — Event was placed on hold due to an internal error in Zeke processing.

Heading	Description	
	NET	Event was placed on hold due to a networking error.
	OPER	Event was placed on hold because a ZHOLD command was issued for the event.
	PSID	The Posid generation option is set to N (see page 566) and the Control field on the EMR is set to Y. With these settings, Zeke has no way to track a remote job, so the event was placed on hold. In order for Zeke to track a remote job, the Posid generation option must be set to Y. Otherwise, Control must be set to N, so that Zeke will not attempt to track the remote job.
	REXX	OASIS ECF error. OASIS encountered an error dispatching the REXX exec for a REXX event. OASIS should display an error message on the system console.
	RFSH	A ZREFRESH command was issued for this event. The event was refreshed and placed on operator hold.
	SEC	Job does not have the authority to run on the platform it was sent to. The event was placed on hold.
	SJCL	Zeke encountered an error reading the JCL while attempting to dispatch the event. The event was placed on hold.
{unlabeled}	The Netregid of the execution system (the TARGET value).	

Event Display Format—INFO

This format is used if the INFO parameter is specified:

```
Z0926I DATE    VERS SCHED TYPE  JOB/EVT-NAME  APPL    GRP USERID
000014 2020329 0001 00:00 JOB   DEP FROM 6K1  CER     CER          * HOLD
000003 2020328 0001 00:00 JOB   NEWTST03      NEW     NEW          T
000003 2020332 0003 00:00 JOB   NEWTST03      NEW     NEW          T
000004*2020332 0001 00:00 JOB   NEWTST04      NEW     NEW          *
000013 2020332 0001 00:00 JOB   TO-TOBIE-NT   CER     CER carlr    * HOLD NET
000008 2020332 0001 00:00 JOB   NEWT          NEW     NEW NEW      T
000009 2020332 0001 00:00 JOB   TO-ZCAL HP    CER     CER richards * HOLD NET
000014 2020332 0001 00:00 JOB   DEP FROM 6K1  CER     CER          T
000015 2020332 0001 00:00 JOB   CERTST        CER     CER          T
000013 2020332 0002 00:00 JOB   TO-TOBIE-NT   CER     CER carlr    * HOLD NET
```

These are the column headings:

Heading	Description
Z0926I	Message ID for the column heading. This column displays the event numbers.
DATE	Julian date of the SQR.
VERS	Version of the event.
SCHED	Normal dispatch time.
TYPE	Event type.
JOB/EVT-NAME	Jobname (for a job event); otherwise, the event name.
APPL	Application ID.
GRP	Group ID.
USERID	User ID.
{unlabeled}	The processing status of the event. See "Event Status/Reason Code" on page 443 for an explanation of each status code. One of the symbols described in "ZDISPLAY Formats" on page 311 is also displayed to the left of the status code.
{unlabeled}	An N indicates the job was submitted with remote system specified as the target. If blank, the job was submitted to run locally.

Event Display Format—JOB

This format is used if the JOB parameter is specified (without the DONE parameter):

```
Z0923I DATE    VERSN  JOB NAME SCHED CLASS  DP TPS STATUS    JOBID    TARGET
000014 2020329 00001  CERJ14   00:00        50               JOB05119
000003 2020328 00001  NEWTST03 00:00        50
000003 2020332 00001  NEWTST03 00:00        50     HOLD OPER
000004*2020332 00001  NEWTST04 00:00        50
000013 2020332 00001  TESTJOB1 00:00        50     HOLD NET           TOBIE
000008 2020332 00001  NEWTST08 00:00        50
000009 2020332 00001  CERJ9    00:00        50     HOLD NET           ZTEAMH
000014 2020332 00001  CERJ14   00:00        50
000015 2020332 00001  CERJ15   00:00        50
000013 2020332 00002  TESTJOB1 00:00        50     HOLD NET           TOBIE
```

These are the column headings:

Heading	Description
Z0923I	Message ID for the column heading. This column displays the event numbers.
DATE	Julian date of the SQR.
VERSN	Version of the event.
JOB NAME	Jobname.
SCHED	Normal dispatch time.
CLASS	Class or class list the event is authorized to run under.
DP	Dispatch priority code.
TPS	Number of tape drives required.
STATUS	The processing status of the event. See "Event Status/Reason Code" on page 443 for an explanation of each status code.
JOBID	JES job ID.
TARGET	Netregid of the remote system the JCL will execute on (if applicable). If the job was submitted to run locally, this field is blank.

Event Display Format—JOB and DONE

This format is used if the JOB and DONE parameters are entered:

```
Z0970I DATE    VERSN JOB NAME SCHED CLASS  DP CCODE STATUS    R JOBID
000011 2020234 00000 NEWJOBA  00:00 A      50 C0000 16:43       JOB00429
000011 2020234 00000 NEWJOBA  00:00 A      50 C0000 16:24       JOB03334
000012 2020101 00001 NEWJOBB  00:00        50 C0000 15:38
000014 2020239 00001 NEWJOBD  00:00        50 C0000          @
000021 2020101 00000 NEW806   00:00        50 S806  18:55       JOB00512
000021 2020105 00000 NEW806   00:00        50 S806  10:59       JOB02180
            RERUN
000031 2020093 00000 NEWXDCB  00:00        50 C0000 19:05       JOB09251
000031 2020123 00000 NEWXDCB  00:00        50 C0000 11:53       JOB00287
000041 2020196 00000 NEWBPXBA 00:00        50 C0000 13:24       JOB02916
000011*2020101 00000 NEWJOBA  00:01 A      50 C0000             JOB00389
000034*2020239 00000 NEWJOBR1 00:01        50 C0000             JOB09639
000012 2020171 00001 NEWJOBB  23:00        50 C0000 17:05
Z0905I NUMBER OF SCHEDULE ENTRIES SELECTED WAS 00013 SYSTEM NEWA
```

These are the column headings:

Heading	Description
Z0970I	Message ID for the column heading. This column displays the event numbers.
DATE	Julian date of the SQR.
VERSN	Version of the event.
JOB NAME	Jobname.
SCHED	Normal dispatch time.
CLASS	Class or class list the event is authorized to run under.
DP	Dispatch priority code.
CCODE	Abend or condition code.
STATUS	The column to the right of CCODE displays the status time.
R	An `N` indicates the job was submitted with remote system specified as the target. If blank, the job was submitted to run locally.
JOBID	JES job ID (if the job has become ACTV (i.e., active) on the system).

Event Display Format—NOTE

This format is used if the NOTE parameter is specified. The screen format is the same as the main display format except the existing Note Documentation is displayed for each event.

```
Z0922I DATE    RDATE    VER TYPE JOB/EVT NAME DP SCHED FREQ  CNT    STATUS
000014 2020329 2020329  001 JOB  CERJ14       50 00:00          1 *  HOLD
000003 2020328 2020328  001 JOB  NEWTST03     50 00:00 00:01  123 T
  NOTE: FROM THE DOCUMENTATION OPTION, YOU CAN DISPLAY EXISTING
        DOCUMENTATION FOR AN EVENT.
        YOU CAN MAINTAIN SCRATCH PAD, NOTE, TEXT, OR DATASET
        DOCUMENTATION IF DOCUMENTATION ALREADY EXISTS FOR THIS
        EVENT.
000004*2020332 2020332  001 JOB  NEWTST04     50 00:00          1 *
000013 2020332 2020332  001 JOB  TESTJOB1     50 00:00          1 *  HOLD NET
  NOTE:  THIS IS A NOTE FOR EVENT NUMBER ONE.
         THIS NOTE CAN BE UP TO
         SIX LINES IN LENGTH.
         FREE-FORM INFORMATION
         CAN BE KEYED IN DESCRIBING
         JOB REQUIREMENTS.
000009 2020332 2020332  001 JOB  CERJ9        50 00:00          1 *  HOLD NET
  NOTE:  EVENT 8 REQUIRES
         THE PAYROLL EDIT
         TO BE IN BALANCE.
000014 2020332 2020332  001 JOB  CERJ14       50 00:00          1 T
```

Event Display Format—WAIT

This format is used if the WAIT parameter is specified:

```
Z0924I DATE    VERS TYPE JOB NAME SCHED CLASS DPRI TPS R REASON          STAT
000014 2020329 0001 JOB  CERJ14   00:00         50       NO DQ ENTRY     HOLD
000004*2020332 0001 JOB  NEWTST04 00:00          1     N NEED OPER OK    HOLD
000013 2020332 0001 JOB  TESTJOB1 00:00         50       NEED RESOURCES  HOLD
000009 2020332 0001 JOB  CERJ9    00:00         50     N TAPE DRIVES REQ HOLD
000013 2020332 0002 JOB  TESTJOB1 00:00          2     N AWAITING RETRY  HOLD
000024 2020332 0001 JOB  TESTJOB2 00:00         50       NOT DUR PGM/JOB
```

Additional details are displayed for events with NOTDURING conditions if the NOTD and/or EXTENDED parameters are included.

Example 1:

```
ZD WAIT EXTENDED
Z0924I DATE    VERS TYPE JOB NAME SCHED CLASS DPRI TPS R REASON          STAT
000041*2020224 0000 MSG  NOTD_TES 00:00         50       NOT DUR PGM/JOB
     waiting on JOB ZEKEQAUT(JOB02436), event 000040 - active on SYSA
Z0905I NUMBER OF SCHEDULE ENTRIES SELECTED WAS 00001 SYSTEM ASGX320A
```

Example 2:

```
ZD WAIT NOTD EXTENDED
Z0924I DATE    VERS TYPE JOB NAME SCHED CLASS DPRI TPS R REASON          STAT
000041*2020224 0000 MSG  NOTD_TES 00:00         50       NOT DUR PGM/JOB
     waiting on PGM ZEKE4QA  - active in ZEKEQAUT(JOB02436), event 000040
   0 WHEN: (NOTDURING PGM ZEKE4QA)
Z0905I NUMBER OF SCHEDULE ENTRIES SELECTED WAS 00001 SYSTEM ASGX320A
```

These are the column headings:

Heading	Description
Z0924I	Message ID for the column heading. This column displays the event numbers.
DATE	Julian date of the SQR.
VERS	Version of the event.
TYPE	Event type.
JOB NAME	Jobname for a job event; otherwise, the event name.
SCHED	Normal dispatch time.
CLASS	Class or class list the event is authorized to run under.
DPRI	Dispatching priority code.
TPS	Number of tape drives required.
R	An `N` indicates the job was submitted with remote system specified as the target. If blank, the job was submitted to run locally.
REASON	The reason the event is waiting in the dispatch queue. See “Event Status/Reason Code” on page 443 for descriptions of the reason codes.
STAT	The processing status of the event. See “Event Status/Reason Code” on page 443 for an explanation of each status code.

Heading	Description

waiting on...	For events with NOTDURING conditions, this heading displays information about the job or program that currently is preventing the event from being dispatched. These are the possible message formats:
	
	`waiting on PGM pgmname – active in jobname(jobid), event nnnnnn`
	
	The specified program is executing in the specified job.
	
	`waiting on JOB jobname, event nnnnnn – Zeke dispatch queue`
	
	The specified job is waiting in the dispatch queue.
	
	`waiting on JOB jobname(jobid), event nnnnnn – JES input queue`
	
	The specified job is waiting in the JES input queue.
	
	`waiting on JOB jobname(jobid), event nnnnnn – active on mvsname`
	
	The specified job is executing on the specified operating system.
	
	`waiting on JOB jobname(jobid), event nnnnnn – active on cpuid`
	
	The specified job is executing on the specified Zeke system.
	
	where:
	
	pgmname is the NOTDURING program name.
	
	jobid is the JES job ID.
	
	jobname is the NOTDURING jobname or the job in which *pgmname* is active.
	
	nnnnnn is the event number associated with *jobname*.
	
	jesname is the JES name of the system where *jobname* is executing.
	
	mvsname is the operating system name.
	
	cpuid is the Zeke CPU ID.

Event Display Format—WHENDETAIL

This format is used if the WHENDETAIL parameter is specified. The screen format is the same as the main display format except for the addition of the version of the WHEN condition that was used for the scheduled event.

```
Z0922I DATE     RDATE  VER TYPE JOB/EVT NAME DP SCHED FREQ  CNT    STATUS
000014 2020329 2020329 001 JOB CERJ14        50 00:00          1 * HOLD
    0 WHEN: (*EOJ CERJ3 AT ZTEAM6K1)
000003 2020328 2020328 001 JOB NEWTST03      50 00:00 00:01  123 T
    0 WHEN: (EOJ JOBA AT A)
000003 2020332 2020332 001 JOB NEWTST03      50 00:00          1 T
    0 WHEN: (EOJ JOBA AT A)
000008 2020332 2020332 001 JOB NEWTST08      50 00:00          1 T
    1 WHEN: (EOJ NEWTST04)
000014 2020332 2020332 001 JOB CERJ14        50 00:00          1 T
    0 WHEN: (EOJ CERJ3 AT ZTEAM6K1)
000015 2020332 2020332 001 JOB CERJ15        50 00:00          1 T
    0 WHEN: (EOJ TESTJOB1 AT ZTEM6K1)
000024 2020332 2020332 001 JOB TESTJOB2      50 00:00          1 T
    0 WHEN: (NOTDURING PGM ZEKE4QA)
```

Event Display Format—XREF

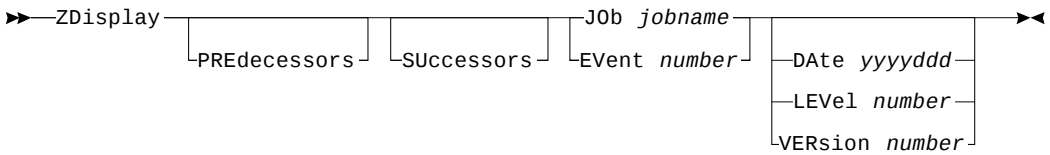
This format is used if the XREF parameter is specified:

```
Z0922I DATE    RDATE   VER  TYPE JOB/EVT NAME DP SCHED FREQ  CNT    STATUS
000003 2020328 2020328 001  JOB  NEWTST03     50 00:00 00:01  123 T
    0 WHEN: (EOJ JOBA AT A)
000003 2020332 2020332 002  JOB  NEWTST03     50 00:00          1 T
    0 WHEN: (EOJ JOBA AT A)
000034 2020336 2020336 001  JOB  KAMTST03     50 00:00          1 T
    0 WHEN: (EOJ JOBA AT A AND WEOJ JOBB)
```

Displaying Preceding and Succeeding Events (PathFinder)

You can display the chain of events or hierarchy of events that are dependent on other events (as specified in the WHEN condition of the EMR for dependent events). See "Event Display Format—SUCCESSOR/PREDECESSOR" on page 334 for a description of the format and headings.

Syntax



Parameters

You can use these parameters with the ZDISPLAY command to display predecessor/successor information:

Parameter	Description

DAte	Displays events with the specified schedule date (*yyyyddd* or *yyddd*). For example, this command displays all events on which event 27 for January 1, 2020 is dependent:
	`ZD PRE EV 27 DA 2020001`
	By issuing the ZDISPLAY command with a date value of `99999`, you can display an event with the schedule date of the earliest event in the schedule that is dependent on the event being displayed. For example, this command displays the event with the earliest schedule date that has not been dispatched because it is waiting on event 234:
	`ZD SU EV 234 DATE 99999`
	Note:
	When using *yyddd* format, if *yy* is greater than or equal to `50`, then the date is considered a twentieth century date (`19`*xx*); if it is less than `50`, it is considered a twenty-first century date (`20`*xx*).

EVent	Displays events that are dependent on or are a dependency for the specified event number.
JOb	Displays events that are dependent on or are a dependency for the specified jobname.
LEVel	Displays the specified number of levels in a particular hierarchy of events. The default level is 1, which displays those events immediately dependent on the specified event. The valid values range from `1` through `999`, or enter an asterisk (`*`) to display all levels. For example, this command displays up to three levels of events that are dependent on event 100: `ZD SU EV 100 LEV 3`
PREdecessors	Displays the events that are prerequisites for the specified event: For example, this command displays all events on which JOB1 is dependent: `ZD PRE JO JOB1` If you specify an event and there are multiple versions of the event in the current schedule, you must use the VER parameter to specify a version.
SUccessors	Displays the events that are dependent on the specified event. For example, this command displays all events that are dependent on event 26: `ZD SU EV 26`

Parameter	Description
VERsion	Specifies a particular version of an event when there are multiple versions of the event in the current schedule.

You can display both predecessor and successor events by specifying both parameters. For example, this command displays all jobs on which PAYROLL1 is dependent and all jobs that are dependent on PAYROLL1:

```
ZD PRE SUC JOB PAYROLL1
```

Event Display Format—SUCCESSOR/PREDECESSOR

This format is used if the SUCCESSOR and/or PREDECESSOR parameters are entered:

```
ZnnnnI PREDECESSORS & SUCCESSORS FOR THE REQUESTED EVENT:
LVL JOB/EVT NAME TYPE EVENT    DATE   VERS WHEN TRIGGER NAME T-VER STATUS AVDUR
  2 VCOM#Y4      VCOM 000001  2020214 0002                          PEND
  2 ZCOM#Y3      ZCOM 000002  2020214 0001                          ACTV
>>1 JOBX2        JOB  000003  2020214 0001 EOE  VCOM#Y4       T           00:25
                                           WEOE ZCOM#Y3
  2 JOBY2        *** NOT IN THE SCHEDULE    -   REMOTE EVENT ***
  2 WORK#Y1      WORK 000005  2020214 0001                          SUCC
>>1 JOBX1        JOB  000006  2020214 0001 EOJ  JOBY2         T           00:20
                                           WEOE WORK#Y1
--------------------------------------------------------------------------------
                                           EOJ  JOBX1
--------------------------------------------------------------------------------
>>1 JOBB1        JOB  000008  2020214 0001 EOJ  JOBA          T    HOLD   00:15
  2 JOBC2        JOB  000010  2020214 0004 WEOJ JOBB1         T           00:00
>>1 JOBB2        JOB  000011  2020214 0001 EOJ  JOBA          T           00:35
  2 ZCOM#C3      ZCOM 000012  2020214 0001 WEOE JOB#B2        T
  2 VCOM#C4      VCOM 000013  2020214 0003 EOE  JOB#B2        T
```

These are the column headings:

Heading	Description
LVL	Indicates the level of hierarchy of the job flow. 0 Original event or job as specified in the ZDISPLAY command. 1 Predecessor or successor of the specified event or job. 2 Predecessor or successor of a level 1 predecessor or successor.
JOB/EVT NAME	Jobname (for a job event); otherwise, the event name.
TYPE	Event type.

Heading	Description
EVENT	Event number.
DATE	Julian date of the SQR.
VERS	Version of the event.
WHEN	Type of WHEN condition.
TRIGGER NAME	Jobname or event that triggers the WHEN condition for the jobname or event listed.
T-VER	Triggering version of the event.
STATUS	The processing status of the event. See "Event Status/Reason Code" on page 443 for an explanation of each status code.
AVDUR	Average duration time for a job or event.
{unlabeled}	If two or more events have the same name and one of the events triggers another, then both events with the same name will appear with the word DUP beside each one in the right margin of the screen.

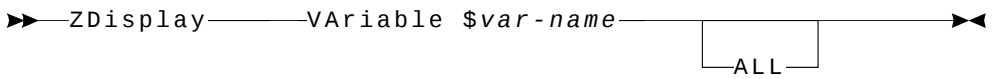
Displaying Variables

You can display the value and current status of a Zeke variable.

Note:

This command displays Zeke variables only; it does not display OASIS variables.

Syntax



Parameters

You can use these parameters with the ZDISPLAY command to display the value of a Zeke variable and related status information:

Parameter	Description
VAriable	Specifies the name of the Zeke variable to display. The Zeke variable must begin with a dollar sign ($) and can be from two to 16 characters long. For example, this command displays the value of Zeke variable $XYZ: `ZD VAR $XYZ`
ALL	Displays the value of the Zeke variable and related status information. For example: `ZD VAR $ABC ALL`

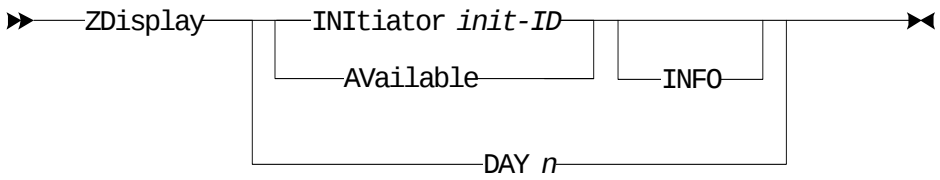
Note:

To display event variables and their values, see "Displaying Event Information" on page 311.

Displaying Initiator Information

You can display the times each initiator is available and job class limits. (You specify the days and times each initiator is available and the job class capacity through the Zeke online facility.)

Syntax



Parameters

You can use these parameters with the ZDISPLAY command to display the desired information:

Parameter	Description
AVailable	Displays the times an initiator is available and the job class limits. All initiators defined on the System Initiator/Partition Directory screen are displayed, unless the INITIATOR parameter is specified. The current day is assumed, unless the DAY parameter is specified. **Note:** If you are using JES3 or have the DispSel generation option set to N, the ZDISPLAY AVAILABLE command is not valid. For example, this command displays today's times and classes for all defined initiators: `ZD AV`

DAY	Display initiator information for the specified day. You specify a value ranging from 1 through 7 to represent the day. The current day is assumed, unless this parameter is used.
	Examples:
	`ZD AV,DAY=5`
	Display initiator information for Friday.
	`ZD DAY=5`
	Display initiator information for Friday (AV is assumed).
	`ZD DAY 4 INI I5`
	Display initiator I5 information for Thursday.

Parameter	Description
INFO	Specify this parameter with the AVAIL or INIT parameter to display this information on the last update and last reload of the current initiator: • System name, or asterisks (********) if the generic initiator was loaded. • JES or Power ID. • Initiator description. • User ID, date, and time the initiator was last loaded into storage. • User ID, date, and time the initiator was last updated on the Zeke database. **Note:** Last update information is available only for Zeke systems Version 6.1 or later. If you have a mixed-release Zekeplex, then the INFO parameter will not display accurate information for a initiator that was last updated on a Zeke 6.0 system.
INItiator	Displays the named initiator information only. For example, this command displays information for initiator I7: `ZD INI I7`

Display Format—AVAILABLE

This format is used if the AVAILABLE parameter is specified:

```
 Z0930I ID CLASSES    START/STOP TIME RANGES FOR MONDAY
-BUSY-   1  AS        00:00-23:59
-IDLE-   2  AS        00:00-23:59
Z099HI        SYSA      *GLOBAL
Z099HI CLASS  CURR MAX  CURR MAX  HELD
Z099II   A      4   10    -    -
Z099II   B      -    -    8   NO   YES
Z099II   C      -    -    -    -   YES
Z099II   D      0    5    4   10       *DUPLICATE*
```

These are the column headings:

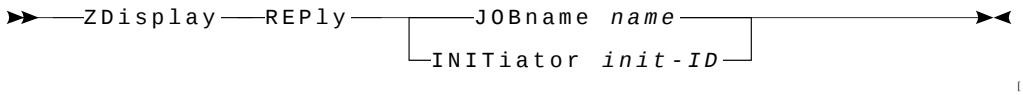
Heading	Description
Z0930I	Message ID for the column headings. The first column displays the initiator status.
ID	Initiator ID.
CLASSES	Classes the initiator can run.

Heading	Description
START/STOP	Starting and ending times the initiator is available to Zeke. There are up to four ranges.
TIME RANGES	Another set of times the initiator is available to Zeke.
FOR MONDAY	Status for the requested day of the week.
Z099HI	Message ID for the column headings. The first line displays the name of the active system.
CLASS	Single-character job class. **Note:** Job classes are displayed only if they have been defined with a maximum capacity or if they are on hold.
CURR	Current number of Zeke-dispatched jobs active or waiting. If the class is defined as unlimited, then NO is displayed. If the class has no capacity defined, a hyphen (-) is displayed.
MAX	Maximum number of Zeke-dispatched jobs allowed for this job class. If the class is defined as unlimited, then NO is displayed. If the class has no capacity defined, a hyphen (-) is displayed.
HELD	Indicates whether the job class has been held (using the operator command ZHOLD JOBCLASS). **Note:** A held class prevents Zeke from submitting jobs to JES for this job class. It does not prevent jobs from running in that class if they are submitted by a source other than Zeke.
DUPLICATE	Indicates whether the job class is defined with maximum capacities for both the local Zeke system and for the Zekeplex (where multiple Zeke systems share a database). **Note:** Only the local job class capacity is considered by Zeke for this system.

Displaying Automatic Reply Information

You can display the active automatic reply elements for a given job event or initiator. If a Zeke job event is running, Zeke displays the messages and replies that are active for that job event.

Syntax



Parameters

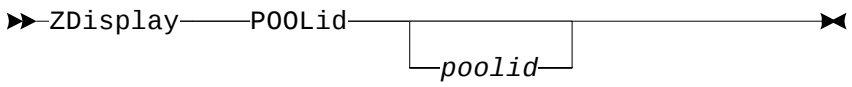
You can use these parameters with the ZDISPLAY command to display the desired information:

Parameter	Description
`INITiator`	Displays the active automatic reply elements for a given initiator. If a Zeke job event is running in the selected initiator, Zeke displays the messages and replies that are active for that job event. For example, this command displays replies for initiator T3. (Issue this command while the job is running and waiting for a reply): `ZD REPLY INIT T3`
`JOBname`	Specifies the name of the job event for which messages and auto replies are to be displayed. For example, this command displays replies for the job event TESTXYZ: `ZD JOB TESTXYZ REPLY`
`REPly`	Requests a display of auto reply information.

Displaying System Pool Information

You can display the system IDs contained in the specified pool (message Z0974I).

Syntax



Parameter

Use this parameter with the ZDISPLAY command to display system pool information:

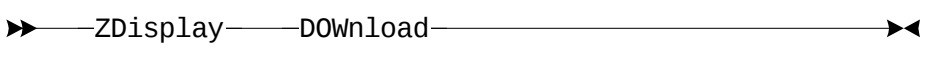
Parameter	Description

`POOLid`	Specifies the pool ID to be displayed (or to display all system pools).
	Examples:
	This command displays the system IDs contained in the pool named POOL1:
	`ZD POOL POOL1`
	This command displays the system IDs contained in all active, in-memory system pools:
	`ZD POOL`

Displaying Zeke Agents

You can display a list of Zeke Agents that are enabled for schedule download (message Z09D1I/Z09D2I/Z09D3I) along with each agent's description, Netregid, download status, and whether the Zeke Agent has been deleted.

Syntax



Parameter

Use this parameter with the ZDISPLAY command to display download agent information:

Parameter	Description
DOWnload	Displays information about schedule download agents and their processing statuses.

Display Format—DOWNLOAD

This format is displayed when the DOWNLOAD parameter is specified:

```
ASG-Zeke              Zeke Command Output Display   BROWSE          Row 1 of 6
Command ===>                                                   Scroll ==> PAGE

Please enter a valid Zeke operator command or option number.
Press PF3/PF15 key to return to the schedule control function panel.

--------------------------------------------------------------------------------
Z09D1I NETREGID  STATUS  SCHEDULE#   DESCRIPTION
Z09D2I NTAGENT   NTSYNC  00:00       ZEKE AGENT NT
Z09D2I OS400AGT  NTSYNC  00:00       ZEKE AGENT OS400
Z09D2I UNIXAGNT  NTSYNC  00:00       ZEKE AGENT UNIX
Z09D2I VMSAGENT  NTSYNC  00:00       ZEKE AGENT VMS
Z09D3I 00004 SCHEDULE DOWNLOAD AGENTS
****************************** Bottom of data ********************************
```

These are the column headings:

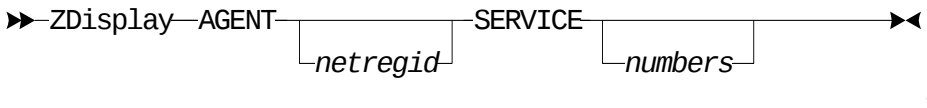
Heading	Description	
Z09D1I	Message ID for the column heading.	
NETREGID	Zeke Agent Netregid	
STATUS	Status of the Zeke Agent. These are the valid statuses:	
	UNKNWN	Zeke Agent status has not been sent to Zeke.
	NTSYNC	Zeke Agent is not synchronized with Zeke.
	IDLE	Zeke Agent is waiting for work.
	DLDING	SQR data is being downloaded to the Zeke Agent.
	WAIT	Zeke has downloaded a schedule or job to the Zeke Agent and is waiting for a response.

Heading	Description	
	OOSYNC	Zeke Agent and Zeke are out of synchronization.
	INACT	Zeke is not active.
	UNSUPP	This Zeke Agent does not support schedule download.
DESCRIPTION	Optional user-defined descriptive text.	
(deleted)	This Zeke Agent has been removed from the download agent table in the Zeke database, but might still have work in progress.	

Displaying Remote Agents

You can display the remote Zeke and Zena agents that are known to Zeke, and are qualified to process remote events or satisfy remote triggers.See the *ASG-Zeke Scheduling for z/OS User's Guide* for more information on how remote agents are eligible for remote processing.

Syntax



Parameters

Use these parameters with the ZDISPLAY command to display remote agents:

Parameter	Description
AGENT	Displays all remote agents currently known to Zeke.
netregid	Displays remote agents that process any service for the specified Netregid.
SERVICE	Displays remote agents that process any of the specified input service numbers.
numbers	Displays the remote agents that process the specified input service number(s).

Display Format—AGENT

This format is displayed when the AGENT parameter is specified:

```
Z9B63I     AGENT                                                  COUNT  RETRY
 DMS       ASGSYSA
           SERVICES:  ALL
           NETREGIDS: ASGSYSA
 ZenaAgnt  znagent.10.21.0.91.7913                                  0      0
           SERVICES:  110 207 214 5 225 241
           NETREGIDS: ZTEAM64
 ZenaAgnt  znagent.10.21.0.85.7913                                  0      0
           SERVICES:  5 241 214 110 225 207
           NETREGIDS: ASGZENA
 ZekeSvc   zekesvcs.ASGPLEX.SYSA                                    0      0
           SERVICES:  128 129
           NETREGIDS: ASGPLEX ZKSERVA1 ZKSERVA2 ZKSERVA3
 ZekeSvc   zekesvcs.ASGZKPLX.SYSB                                   0      0
           SERVICES:  128 129
           NETREGIDS: ASGZKPLX ZEKSYSB1 ZEKSYSB2 ZEKSYSB3 ZEKSYSB4
```

These are the column headings and informational fields:

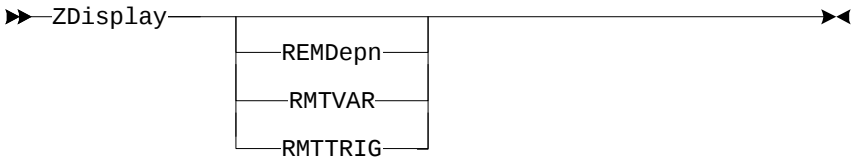
Heading	Description	
Z9B63I	Message ID.	
AGENT	Agent type. These are the possible types:	
	`DMS`	Remote Zeke Agent registered through OASIS/DMS by the Netregid value specified in the Zeke generation options.
	`ZekeSvc`	Remote Zeke Services agent registered through RI Server by the agent ID as defined in the ZENVIRN environment file. By default, the ID includes the *`plexid.sysid`*.
	`ZenaAgnt`	Remote Zena agent registered through RI Server.
SERVICES	Input service numbers processed by the agent. For a Zeke Services agent, these are the supported service numbers:	
	`128`	Zeke job
	`129`	Zeke remote trigger
	For a Zena agent, see the *ASG-Zena Installation Guide* for supported services.	

Heading	Description
NETREGIDS	Netregids under which the agent processes the specified services. • For a DMS agent, the Netregid value specified in the Zeke generation options for the Zeke Services is displayed. • For a Zeke Services agent, the plex ID is displayed along with any Netregids specified in the Zekeplex Netregids List. • For a Zekeplex, which may have multiple Zeke Services agents, all Netregids specified in the Zekeplex Netregids List are displayed. • For a Zena agent, the Netregid specified in the Zena agent configuration file is displayed. See the *ASG-Zena Installation Guide* for details. **Note:** See the *ASG-Zeke Scheduling for z/OS User's Guide* for information on managing the Zekeplex Netregids list.
COUNT	(Valid only for remote events sent to Zena agents.) Number of remote events Zeke has dispatched to the agent.
RETRY	(Valid only for remote events sent to Zena agents.) Number of times Zeke has attempted to dispatch a remote event to the agent and experienced a network failure. When the RETRY count reaches 10, the agent is flagged as disabled and Zeke no longer dispatches events to the agent until it re-registers with RI Server.

Displaying Remote Events, Variables and Triggers

You can display information about the remote events, dependencies and triggers, and variables that are submitted to the Zekeplex from another system for execution. The output indicates the originating system.

Syntax



Parameters

These parameters are valid with the ZDISPLAY command to display remote processing details:

Parameter	Description
REMDepn	Displays information about remote dependencies.
RMTTRIG	Displays the current remote triggers in the Zekeplex.
RMTEVT	Selects the remote events that have been sent to the Zekeplex for execution.
RMTVAR	Displays the remote originator variables that have been sent to this Zeke system (i.e., the system where you issue the command). **Note:** Zena sends its variables to every Zeke in the Zekeplex. You can issue the ZD RMTVAR command on each Zeke system to verify whether it contains the same set of originator variables from Zena. If necessary, re-cycle (COLD or TRACK) each Zeke that is out of sync in order to bring its set of originator variables into accordance with the Zekeplex.

Display Format—RMTEVT

This format is displayed when the RMTEVT parameter is specified:

```
Z092FI F79B53A264A8 0000000407 0000000001 BAA8805034E9 011696
2020/01/01 20:11
Z092FI F79B53A264A8 0000000408 0000000001 C2B7C2121757 011697
2020/01/01 20:15
Z092FI Displayed 000002 remote events
```

These are the column headings and informational fields:

Heading	Description
Z092FI	Message ID.
Originator	Originating system that submitted the event to this Zekeplex.
Process	Process ID of the task in the originating system.
Item	Item ID of the task in the originating system.
Run	Run ID of the task in the originating system.
Event	Event number assigned to the remote event by the Zekeplex.

Heading	Description
yyyy/mm/dd hh:mm	Date and time that the remote event was received by the Zekeplex.

Display Format—RMTTRIG

This format is displayed when the RMTTRIG parameter is specified:

```
Z092GI Originator   Process    Item       Type     Name
Z092GI F79B53A264A8 0000000000 0000001211 AEOE     JOBA
Z092GI F79B53A264A8 0000000000 0000001232 DSN      DATASET1
Z092GI Displayed 000002 remote triggers
```

These are the column headings and informational fields:

Heading	Description
Z09RGI	Message ID.
Originator	Originating system that submitted the trigger to the Zekeplex.
Process	Process ID of the trigger in the originating system.
Item	Item ID of the trigger in the originating system.
Type	Trigger type.
Name	Trigger name (first 45 characters are displayed).

Display Format—RMTVAR

This format is displayed when the RMTVAR parameter is specified:

```
Z092HI Originator   Variable         Value
Z092HI F79B53A264A8 $GLOBALBOOL      TRUE
Z092HI F79B53A264A8 $GLOBALDATE      2020.01.01
Z092HI F79B53A264A8 $GLOBALNUM       999
Z092HI Displayed 000003 remote variables
```

These are the column headings and informational fields:

Heading	Description
Z09RHI	Message ID.
Originator	Originating system that submitted the variable to the Zekeplex.

Heading	Description
Variable	Variable name.
Value	Variable value. (first 59 characters are displayed).

Displaying Communication Record Information

You can display the system IDs that are sharing the Zeke database (message Z0985I/Z0986I). The ZDISPLAY COM command is valid only in multisystem environments.

Syntax



Parameter

Use this parameter with the ZDISPLAY command to display active systems:

Parameter	Description
COM	Displays the system IDs currently active against this database.

If the MultSys generation option is set to N and there is only one Zeke active on the database, Zeke displays this message in response to the ZD COM command:

```
Z0988I  ZEKE MULTI SYSTEM COMMUNICATION IS NOT ACTIVE
```

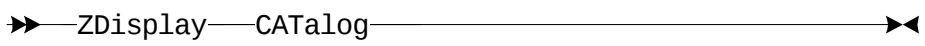
Note:

You can remove an invalid or inactive system ID registration by using the batch utility option CPUDEL (see "OPTIONS—Perform Administrative Tasks" on page 78 for details).

Displaying Database Information

You can display both vault and primary database information (message Z09AKI).

Syntax



Parameter

Use this parameter of the ZDISPLAY command to display database information:

Parameter	Description
CATalog	Displays vault/primary database information.

Displaying Generation Options

You can display information about the local and global generation options that are currently active (i.e., loaded in memory).

Syntax



Parameters

You can use these parameters with the ZDISPLAY command to display GENOPT information:

Parameter	Description
GENopts	Displays a brief description and last modification information for the currently active local GENOPT and the *GLOBAL GENOPT. By default, any updates made to option fields that are pending reload also are displayed (i.e., the value in memory differs from the value in the Zeke database).
ALL	Optional. Displays the values of all generation option fields currently in memory.

Display Format—GENOPT

This format is displayed when the GENOPT parameter is specified:

```
Z9B51I In-memory GENOPT values, loaded by *ZEKEUP* 09/20/2020 14:05:47
Z9B52I   GENOPT name:   ********
Z9B56I     Description: Default local system options
Z9B57I     Last update: ALICE    09/20/2020 17:15:17
Z9B52I   GENOPT name:   *GLOBAL
Z9B56I     Description: Zekeplex global options
Z9B57I     Last update: BOB      09/19/2020 17:30:35
Z9B54W Values marked with * differ from the database copy of *GLOBAL/********
Z9B53I Option    Value             Description
Z9B53I Abhold   *Y                 Yes to hold recurring events if abended
```

These are the column headings and display fields:

Heading	Description
GENOPT name	Name of the active GENOPT (i.e., `********` or `*GLOBAL`).
Description	Description of the GENOPT.
Last update	Date and time the GENOPT was last reloaded (and the user ID that reloaded the GENOPT) or updated (and the user ID or batch jobname that made the update).
Option	Name of the option field contained in the GENOPT or that has been updated and is pending reload.
Value	Current value of the option field contained in the GENOPT or the new value (if the option field has been updated). **Note:** An asterisk (*) indicates whether the value differs from the value currently in memory and is pending reload.
Description	Description of the option field contained in the GENOPT or that has been updated.

Displaying Tracing Messages

You can display the description and status of a trace point as well as manage the tracing settings for many Zeke function calls by activating or deactivating trace points in Zeke.

When a trace point is activated, the output is written both to the data space log and to the system or job log via WTO.

Caution! Because active trace points can generate a large volume of messages in multiple address spaces (as well as require additional CPU and storage), ASG recommends that you do not activate trace points unless you are directed by ASG Customer Support.

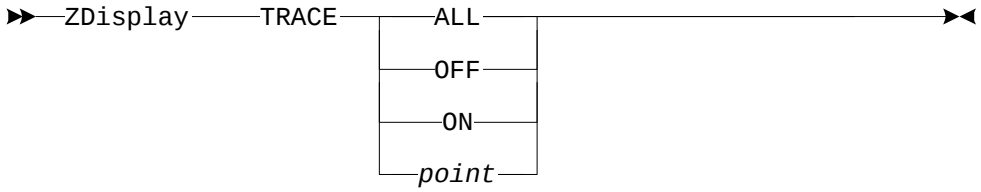
Note:

For information on how to control tracing for external security, see the section on the SET ESITRACE command in the *ASG-OASIS for z/OS Reference Guide*.

Displaying Trace Point Status

You can use ZD TRACE to display the status of active, inactive, or all trace points, or a specific point.

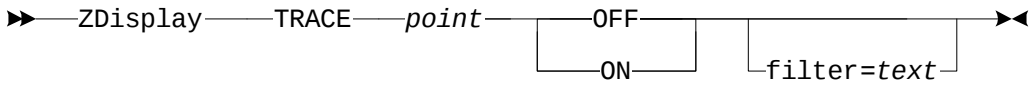
Syntax



Activating Trace Points

You can use ZD TRACE to set a specific trace point as active or inactive. Some trace points enable you to filter the data before it is reported.

Syntax



Parameters

You can use these parameters with the ZDISPLAY TRACE command:

Parameter	Description

TRACE	Controls the display of trace messages and the status of trace points. These are the valid keywords for the ZD TRACE command:
	
	ALL Displays the status of all trace points. For example:
	`ZD TRACE ALL`
	
	OFF Displays the status of trace points that are inactive (i.e., OFF) or deactivates the specified trace point.
	For example:
	• To display the status of inactive trace points, enter this command:
	`ZD TRACE OFF`
	• To deactivate the trace point related to calls to JCL exits, enter this command:
	`ZD TRACE JCLEXIT OFF`
	
	ON Displays the status of trace points that are active (i.e., ON) or activates the specified trace point.
	For example:
	• To display the status of active trace points, enter this command:
	`ZD TRACE ON`
	• To activate the trace point related to calls to JCL exits, enter this command:
	`ZD TRACE JCLEXIT ON`

Parameter	Description

point	Specifies the trace point that you want to activate or deactivate, or for which you want to display its status. These are the valid trace points:
	
	COMM For example:
	`ZD TRACE COMM`
	These commands also are supported for managing this trace point:
	`ZD SCHD ERR`
	`ZD SCHD OFF`
	
	DOWN For example:
	`ZD TRACE DOWN`
	Note:
	Only the first two lines of each message are displayed for this trace point.
	
	JCLExit For example:
	`ZD TRACE JCLEXIT`
	These commands also are supported for managing this trace point:
	`ZD JCLEXIT ON`
	`ZD JCLEXIT OFF`
	
	JCLWto For example:
	`ZD TRACE JCLWTO`
	These commands also are supported for managing this trace point:
	`ZD JCLWTO ON`
	`ZD JCLWTO OFF`
	
	JIJO For example:
	`ZD TRACE JIJO`

Parameter	Description	

	NOTD	For example:
		`ZD TRACE NOTD`
		Note:
		The ZDISPLAY command enables you to view events that are in a NOTDURING wait status (including details about the job or program that is preventing an event from being dispatched). For example:
		`ZD WAIT NOTD EX`
		See "Displaying Event Information" on page 311 and "Event Display Format—WAIT" on page 329 for more information.

	PLEXY	ZEKE05F resource from XCF.
	PLEXZ	ZEKE03E resource from XCF.

	SCHDALL	For example:
		`ZD TRACE SCHDALL`
		These commands also are supported for managing this trace point:
		`ZD SCHD ALL`
		`ZD SCHD OFF`

	SCHDERR	For example:
		`ZD TRACE SCHDERR`
		These commands also are supported for managing this trace point:
		`ZD SCHD ALL ERR NOCOM`
		`ZD SCHD OFF`

Parameter	Description

	SECALL For example:
	`ZD TRACE SECALL`
	Caution! Because this subparameter can display up to 11 lines of messages at each trace point, ASG recommends you use it only when necessary.
	These commands also are supported for managing this trace point:
	`ZD SECEXIT ON ALL`
	`ZD SECEXIT OFF`

	SEcexit For example:
	`ZD TRACE SECEXIT`
	These commands also are supported for managing this trace point:
	`ZD SECEXIT ON`
	`ZD SECEXIT OFF`

	SMFMSG For example:
	`ZD TRACE SMFMSG`
	Note:
	You also can use the ZALTER SMF command to toggle on/off the trace points related to SMF exits.
	This command also is supported for managing this trace point:
	`ZALTER SMFM`

	UCBScan For example:
	`ZD TRACE UCBSCAN`
	These commands also are supported for managing this trace point:
	`ZD UCBSCAN ON`
	`ZD UCBSCAN OFF`

Parameter	Description	
	U83	You must specify a filter to select the datasets being closed for which Zeke issues messages. For example: `ZD TRACE U83 ON DSN fragment` where *fragment* is a full or partial dataset name.

	VARALL	For example:
		`ZD TRACE VARALL`
		These commands also are supported for managing this trace point:
		`ZD DATANAME ALL`
		`ZD DATANAME OFF`

	VARERR	For example:
		`ZD TRACE VARERR`
		These commands also are supported for managing this trace point:
		`ZD DATANAME ERR`
		`ZD DATANAME OFF`

	ZPLEX*c*	Controls tracing of Zeke use of XCF message services (i.e., PLEXCOMM and PLEXNOTD). For example: `ZD TRACE ZPLEXc` where *c* is a single character code (ranging from A through W) that indicates the specific trace point. These are the valid codes: A ZEKE48A1 work check B ZEKE48A1 work items C ZEKE48A1 broadcast D ZEKE48A1 Zeke change E ZEKE 48A1 system change F ZEKE48A1 cleanup G ZEKE 48A1 initialization

Parameter	Description	
	H	ZEKE48A1 polling
	I	ZEKE01A R3REC XCF check
	J	ZEKE05F COMM entries
	K	ZEKE05F work items
	L	ZEKE05F cleanup
	M	ZEKE01A new COMM entry
	N	ZEKE05F broadcast
	O	ZEKE01A COMM entry from XCF
	P	ZEKE05F SQR from XCF
	Q	ZEKE03A SQR from XCF
	R	ZEKE03A CMQ entries
	S	ZEKE05F incoming ZXC detail
	T	ZEKE01A outgoing ZXC detail
	U	ZEKE03A weak satisfy
	V	ZEKE03A CHKETT unnecessary read
	W	ZEKE03A SATLIST detail
	X	Cross-system JCL errors

	These commands also are supported for managing this trace point:
	`ZPLEX DEBUG c ON`
	`ZPLEX DEBUG c OFF`	

Display Format—TRACE

This format is displayed when the TRACE parameter is specified:

```
Z09BBI Trace     Act  Description
Z09BBI --------  ---  ---------------------------------------
Z09BBI NOTD      YES  NOTDURING processing
Z09BBI SCHDERR   YES  Scheduling - only errors
Z09BBI UCBSCAN   YES  UCB scanning
Z09BBI U83       YES  IEFU83 dataset filter
Z09BEI   DSN=MY.DSN.PART
Z09BBI VARERR    YES  Variables - only errors
```

These are the column headings:

Heading	Description	
Z09BBI	Message ID.	
Trace	Trace point name. These are the valid names:	
	`COMM`	Database communication records.
	`DOWN`	Downloaded schedule processing.
	`JCLEXIT`	Calls to JCL exits.
	`JCLWTO`	JCL exit processing.
	`JIJO`	JI/JO toleration processing.
	`NOTD`	NOTDURING evaluation.
	`SCHDALL`	All schedule processing.
	`SCHDERR`	Only error-related schedule processing.
	`SECALL`	All security exit processing.
	`SECEXIT`	Basic security exit processing.
	`SMFMSG`	SMF exit processing.
	`UCBSCAN`	UCB scanning.
	`U83`	IEFU83 SMF exit processing.
	`VARALL`	All variables (dataname) processing.
	`VARERR`	Only error-related variables (dataname) processing.

Heading	Description	
	ZPLEXc	XCF message processing.
Act	Trace point status. These are the valid statuses:	
	-	Trace point is not active (i.e., OFF) for this trace point.
	YES	Tracing is active (i.e., ON) for this trace point.
Description	Trace point description.	

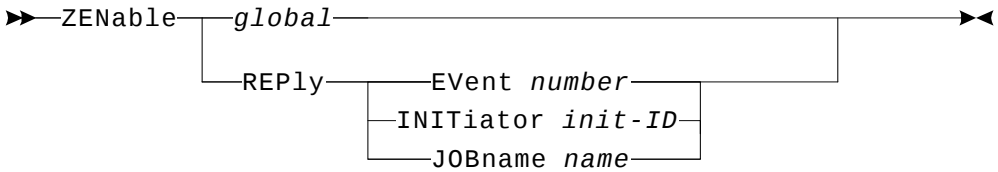
ZENABLE Command

The ZENABLE command reactivates or enables events that have been disabled using the ZDISABLE command. You also can use this to reactivate the automatic reply elements for an event for which automatic replies have been disabled.

Typically, a disabled event that was scheduled for a previous day is dropped by the cu rrent day's first schedule update.

See the *ASG-Zeke Scheduling for z/OS User's Guide* for information on enabling auto replies.

Syntax



Parameters

Use any of the global event selection parameters (starting on page 261) to select events to be enabled. These parameters also can be used with the ZENABLE command:

Parameter	Description
EVent	Event number for which auto replies are to be enabled. This command only has effect if the ZDISABLE command was previously used to disable auto replies for the event. For example, this command enables previously disabled auto-replies for event 55: `ZEN EV 55 REP`
INITiator	Initiator ID to be enabled. This command only has effect if the ZDISABLE command was previously used to disable auto replies for the initiator. `ZEN REP INIT I3` This enables previously disabled auto replies for the job event running in I3.
JOBname	Jobname for which auto replies are to be enabled. For example, this command enables previously disabled job event TS010000: `ZEN JOB TS010000`
REPly	Enables the auto reply for the specified event.

ZHOLD Command

Use the ZHOLD command to place these types of holds on Zeke processing:

- An operator hold on events.
- An operator hold on a initiator.
- An operator hold on a job class.
- A system hold.

Placing Events on Hold

An event hold prevents Zeke from dispatching the specified event(s). The event can be either in the schedule queue or in the dispatch queue when the hold is placed. If the event is in the schedule queue, Zeke moves the event to the dispatch queue when the time and prerequisite requirements are satisfied, but does not dispatch the event.

You can issue this command to display all events currently on hold:

```
ZD HOLD
```

You can issue this command to display events being held for a specified reason:

```
ZD HOLD=reason
```

An event hold remains in effect until it is released specifically with the ZRELEASE command, regardless of any system IPLs. (See “ZRELEASE Command” on page 384.)

Placing an Initiator on Hold

This prevents Zeke from dispatching any jobs to that initiator. It does not prevent jobs from running in that initiator if they are submitted by a source other than Zeke. Only an initiator defined to the Zeke GENSYS option can be held.

You can issue the command `ZD AV` to display initiators currently on hold:

An initiator hold remains in effect until a ZRELEASE command is issued, the system is IPL’d, or Zeke is cycled. (See “ZRELEASE Command” on page 384.)

Placing a Job Class on Hold

This prevents Zeke from submitting jobs to JES for a particular job class. It does not prevent jobs from running in that class if they are submitted by a source other than Zeke.

You can issue the command `ZD AV` to display the job classes currently on hold.

A job class hold remains in effect until a ZRELEASE command is issued or Zeke is cycled. (See "ZRELEASE Command" on page 384.)

Placing a System on Hold

This prevents the local Zeke system from dispatching events until the system hold is released entirely. If particular events also are on operator hold (see "Operator Holds" on page 384), those events can be released individually while the rest of the system is on hold.

Zeke continues to perform its monitoring duties while on hold. Events continue to move from the schedule queue to the dispatch queue as their time and WHEN prerequisites are satisfied, but no events are dispatched until they are released from hold.

You must issue the ZHOLD command from the system to be released. You cannot use the command to release another system.

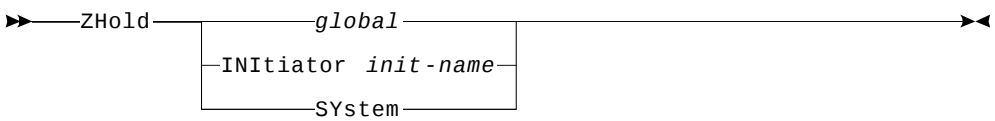
You can enter this command to verify whether there already is a system hold in effect:

```
ZHOLD SYSTEM
```

If there is already a system hold in effect, the command is rejected. If no hold is in effect, the command is accepted and the appropriate confirmation message is displayed.

A system hold remains in effect until a ZRELEASE SYSTEM command is issued, the system is IPL'd, or Zeke is cycled. (See "ZRELEASE Command" on page 384.)

Syntax



Parameters

To place a hold on one or more events, enter one or more global event selection parameters (see the listing beginning on page 261) to select the events. You can use these parameters with the ZHOLD command:

Parameter	Description

INItiator	Places a hold on the specified initiator. When
	A hold onan initiator is not hold and not an event hold or a system hold. In this case, all other hold parameters are ignored.
	Examples:
	This command places an operator hold on initiator A:
	`ZH INI A`
	This command places an operator hold on initiator 10:
	`ZH INI 10`

JOBCLass	Suspends job event dispatching for the specified job class. For example, this command places an operator hold on dispatching for job class A: `ZH JOBCL A` **Note:** The JOBCLASS and CLASS parameters provide different results: `ZHOLD CL A` places an operator hold on all job events that have class `A` specified in their class list. `ZHOLD JOBCL A` places an operator hold on all job submittals to JES for job class `A`.

SYstem	Suspends all event dispatching on the system from which the command is issued. For example:
	`ZHOLD SYS`
	You do not include a system ID with this parameter. If you do so, the system ID is ignored. You cannot use issue this command from one system in order to place another system on hold.For example, if you issue this command from CPUB:
	`ZH SYS CPUA`
	The result is that system CPUB is placed on hold instead of CPUA.
	Any other selection parameters you specify are ignored.

Examples

These are some examples of using global event selection parameters with ZHOLD:

This command places an operator hold on event 45:

```
ZH EV 45
```

This command places an operator hold on any job event that has a jobname beginning with PR:

```
ZH JOB *PR
```

This command places an operator hold on every event in the schedule queue:

```
ZH ALL
```

This command places an operator hold on all job events:

```
ZH JOB *
```

This command places an operator hold on all events with the group name PAY:

```
ZH GRO PAY
```

This command places an operator hold on all events with the application name JOB:

```
ZH AP JOB
```

ZID Command

The ZID command displays this information on the system console:

- Release levels of this Zeke and OASIS system.
- CPU model and serial number.
- Number of remaining days in the 45-day grace period that Zeke will run without a valid license key.
- Current time and date.
- CPU ID and CPU name.
- Set of generation options (GENOPT) currently in use.
- Subsystem ID.
- Plex ID of the Zekeplex in which this Zeke is running.
- System hold state.
- Current value of the Posid generation option (where `P` indicates `Yes` and blank indicates `No`). (See "Posid" on page 566.)
- Netregid for OASIS/DMS services.
- Schedule number used for tracking and synchronizing schedules downloaded to Zeke Agent.
- Database ID.
- Serial number of the volume and name of the dataset that contain the primary Zeke database.
- Name of the dataset and volume that contain the vault, if a vault has been specified.
- Current status of the vault (i.e., active or disabled), if a vault has been specified.

Syntax



Sample Output

```
Z0903I  ZEKE Z620A001 X320A001 2094-07BE2B DAYS: 1 SYSHOLD:NO
Z0903I  TIME 18:50:36 01/27/2020 SATURDAY 2020027
Z0903I  CPUID:P CPUNAME:SYSD     GENOPT:SYSD     SCHED#:0000000012
Z0903I  SUBSYS:ABCD PLEXID:ABCDPLEX CATID:BF4C4077 P NETREGID:SYSD
Z09AKI  ZEKECAT  BF4C4077 VOL=ZEKE99 DSN=ZEKE.TEST.DATABASE
```

ZINFO Command

The ZINFO command displays this information on the system console:

- Release levels of this Zeke and OASIS system.
- CPU model and serial number.
- Number of remaining days in the 45-day grace period that Zeke will run without a valid license key.
- Current time and date.
- CPU ID and CPU name.
- Set of generation options (GENOPTs) currently in use.
- Subsystem ID.
- Plex ID of the Zekeplex where this Zeke is running.
- Database ID.
- System hold state.
- Netregid for OASIS/DMS services.
- Schedule number used for tracking and synchronizing schedules downloaded to Zeke Agent
- Storage statistics for the Zeke schedule tables (e.g., schedule queue tables, dispatch queue tables, dataset name tables, event trigger tables, etc.).

Syntax



Sample Output

```
Z0903I  ZEKE Z620A000 X320A000 2817-0946B6 Days: 1 SYSHOLD:NO
Z0903I  Time 11:17:42 07/17/2020 Tuesday   2020199
Z0903I  Cpuid:P Cpuname:SYSD     Genopt:********
Z0903I  Subsys:ABCD Plexid:ABCDPLEX Catid:C7FF1C67 P Netregid:SYSD
Z0903I  Schd#:00000001 Comm#:00000000 Weak#:00000000
Z0903I           OVERFLOW    # BLKS     # ENT        ALLOC
Z0903I  ZSQT       39,188         2     1,533      392,488
Z0903I  ZDQT        2,900         1        40        2,900
Z0903I  ZPDQ            0         0         0            0
Z0903I  ZCMQ          980         1        40          980
Z0903I  ZRTT            0         0         0            0
Z0903I  ZDST        2,580         1        40        2,580
Z0903I  ZDNT        1,780         1        40        1,780
Z0903I  ZEET        7,060         1     1,100       70,420
Z0903I  ZEJT        6,740         1     1,050       67,220
Z0903I  ZETT        2,580         1        40        2,580
```

This table explains the headings in the output that precede the storage statistics for the various Zeke schedule tables:

Heading	Description
OVERFLOW	Minimum size (in bytes) of an overflow storage block for the table. When you allocate a new block, this field indicates the block size. On startup, Zeke automatically allocates one block per table, then allocates additional blocks as needed. If the amount of storage required to hold the initial table data is *larger* than the OVERFLOW value, the size of the initially allocated block equals the actual amount of storage required for the data. If the amount of storage required to hold the initial table data is *smaller* than the OVERFLOW value, then the size of the initially allocated block equals the OVERFLOW value. If the OVERFLOW and ALLOC sizes are the same, then the minimum size block was large enough to hold all of the initial entries for the table.
# BLKS	Number of blocks of storage allocated to the table. On startup, Zeke automatically allocates one block per table, then allocates additional blocks as needed.
# ENT	Total capacity of the table (in number of entries). If a table needs to contain more entries than indicated by this value, you must allocate additional storage blocks.
ALLOC	Amount of storage (in bytes) allocated to the table.

ZKILL Command

The ZKILL command can be used to terminate Zeke in any of these ways:

Caution! Terminate Zeke's time monitoring, event dispatching, and command processing functions (COLD).ZKILL COLD is equivalent to using the MVS command STOP (`P startedtaskname`).Do not use the MVS command CANCEL to terminate Zeke unless at the direction of ASG Customer Support.Terminate Zeke dispatching only (WARM).Terminate the Zeke system and place it in SMF recording mode (TRACK).ZKILL TRACK enables many system activities to be tracked while Zeke and OASIS are shut down, for example, when you need to apply Zeke or OASIS maintenance, or recover database services by switching to a vault database. With continuous job tracking, disruption to Zeke job dispatching is minimized.Do not use ZKILL TRACK under these conditions:

- If Zeke is restarted on a different database than it is currently using.
- If the database is restored before Zeke is restarted.
- If a full schedule run will occur before Zeke is restarted.
- If you are upgrading to a different release of Zeke.

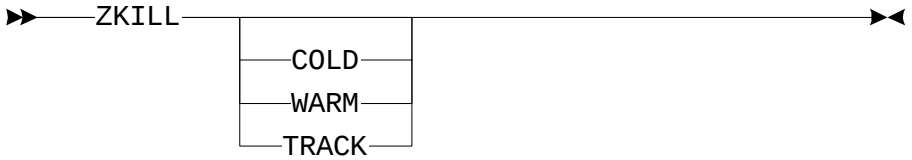
See the *ASG-Zeke Scheduling for z/OS User's Guide* for details about continuous job tracking through SMF recording.

Note:

When you upgrade to a different PTF level under the same release, some PTFs may require you to terminate Zeke (either before or after applying maintenance). For these PTFs, the PTF instructions will indicate whether you can use ZKILL TRACK or WARM instead of ZKILL COLD.

If the ZKILL COLD or TRACK command is issued while any job event is waiting for communications (as indicated by message Z8T01I), Zeke posts the job and enables it to run. In the case of a ZKILL TRACK command, Zeke also tracks the job.See the *ASG-Zeke Scheduling for z/OS User's Guide* for more information on restarting or terminating Zeke.

Syntax



Parameters

One of these parameters is required with the ZKILL command:

Parameter	Description
COLD	Terminates all Zeke processing and releases all Zeke program and table storage. Other 'Z' products within the same address space remain active. (This is equivalent to using the MVS command STOP.) `ZKILL COLD` **Note:** If ZKILL COLD is issued while any job event is waiting for communications, Zeke posts the job to enable it to run. **Note:** To terminate OASIS, you must also issue the XKILL command. ZKILL COLD and XKILL are typically issued to apply maintenance to Zeke and/or OASIS.
TRACK	Terminates Zeke in the same manner as ZKILL COLD, but keeps Zeke's SMF exits active and places Zeke in SMF recording mode. `ZKILL TRACK` After the SMF exits are placed in logging mode, Zeke issues an informational message indicating that system activity is being recorded. Termination continues as it would for a traditional ZKILL COLD, except that only the IEFUJV SMF exit is de-installed. See the *ASG-Zeke Scheduling for z/OS User's Guide* for more information on continuous job tracking through SMF recording, and for the limitations associated with using this option. **Note:** If ZKILL TRACK is issued while any job event is waiting for communications, Zeke posts the job to enable it to run and tracks the job. **Note:** To terminate OASIS after a ZKILL TRACK command is issued, you must also issue the XKILL command.

Parameter	Description
WARM	Terminates Zeke dispatching only. Zeke still performs all tracking of jobs, triggering, and updates. Other products within the same address space remain active, as well as Zeke's SMF exits. `ZKILL WARM` If Zeke is canceled during a cold start, before the schedule has been loaded the first time, or while a schedule reload is in progress, the schedule is freed and Zeke terminates fully. **Note:** To terminate OASIS, you must also issue the XKILL FORCE command. XKILL FORCE, issued along with the ZKILL WARM command, has the same effect as issuing a ZKILL COLD command.
FORCE	Use only if directed to do so by Zeke through an operator console message.

ZMAP Command

The ZMAP command maps the MVS initiators that are running active jobs. This information is provided:

- Current jobname
- Phase name
- Zeke step name
- Zeke event number
- Start time of each job
- Elapsed real time for each job
- CPU time
- Start I/O count

Syntax



Parameters

You can use these parameters with the ZMAP command:

Parameter	Description
none	Displays only the initiators running active Zeke jobs. For example: `ZMAP`
ALL	Displays initiators running active Zeke jobs *and* non-Zeke jobs. Only jobs that started *after* Zeke came up are listed. For example: `ZMAP ALL`

ZOK Command

The ZOK command notifies Zeke when an event receives an operator approval for dispatching. Approval is required for an event if the EMR field Oper OK is set to Y.

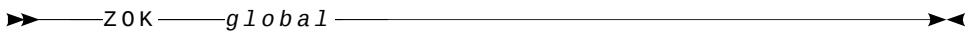
When events that require an operator approval move from the schedule queue to the dispatch queue, the operator is notified by a console message. The events remain in the dispatch queue until the ZOK command is issued.

Note:

For a list of all events that require an operator OK, issue this command:

```
ZDISPLAY NEEDOK
```

Syntax



Parameters

Use the global event selection parameters (see the list beginning on page 261) with ZOK.

Note:

If an event is in the schedule multiple times, use of the ZOK command affects *all occurrences* that match the specified global event selection parameters.

Examples

These are examples of using the ZOK operator command with some of the global event selection parameters.

This command selects event 62 for OPEROK:

```
ZOK EV 62
```

This command selects all job events having a jobname beginning with PR for OPEROK:

```
ZOK JOB *PR
```

This command provides OPEROK for all events that require it:

```
ZOK ALL
```

This command provides OPEROK for all ZCOM events in the dispatch queue:

```
ZOK DQ,ZCOM
```

This command provides OPEROK for all events with the group ID PAY:

```
ZOK GR PAY
```

ZPLEX Command

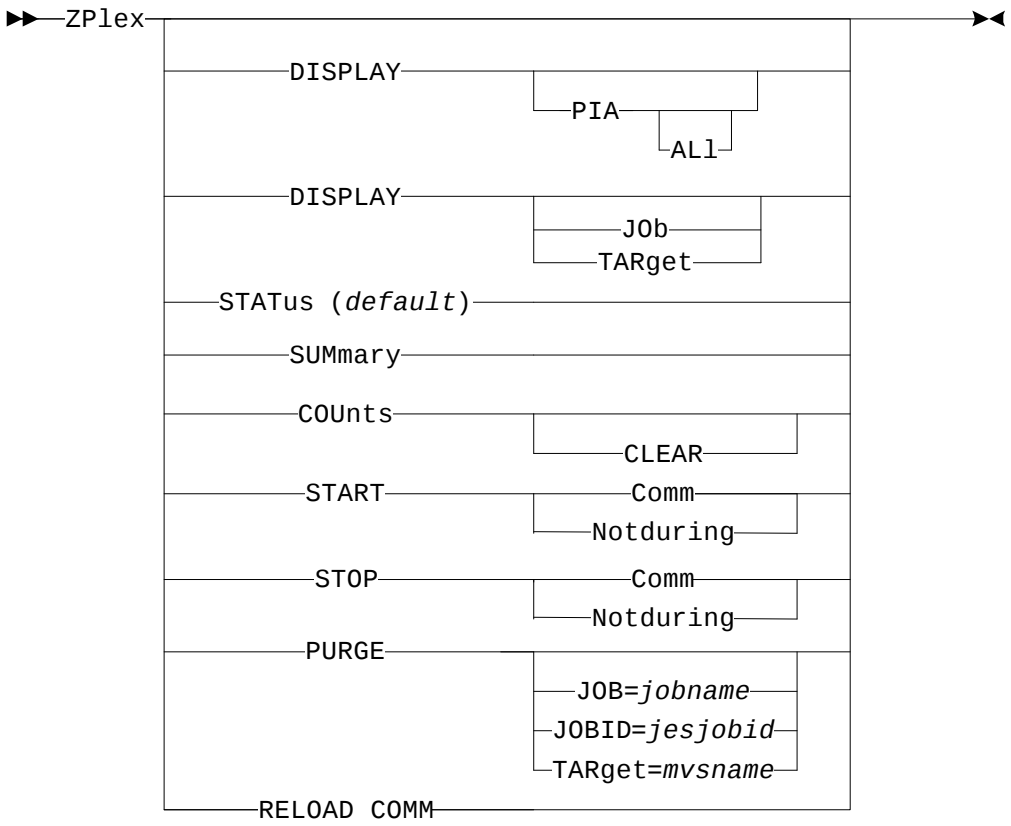
The ZPLEX command is used to manage Zekeplex services. The ZPLEX command can be used for these purposes:

- Display status information about all z/OS systems and Zeke systems in the Zekeplex.
- Display summary information about this Zeke's XCF processing status.
- Display information in Zeke's NOTDURING tables.
- Display SQR statistics (e.g., message sizes, counts, duplicate entries, etc.).
- Start and stop Zekeplex services.
- Remove an entry from the NOTDURING jobs table (if an error occurs).
- Invoke communications recovery and reload the schedule.

Note:

This command is invalid if you run Zeke on a single system only.

Syntax



Parameters

You can use these parameters with the ZPLEX command:

Parameter	Description
`Display`	Displays information that Zeke uses for its NOTDURING processing. Two types of data tables can be reported—local system or JESplex job-tracking. These are the valid keywords: `PIA` — When PIA is coded, this displays information from the local system tracking table. These are jobs that are active on this MVS system; only jobs that started after Zeke initialization are tracked in this table. This keyword is valid when PIA is requested: `ALl` — Reports both active and inactive table entries. If omitted, only active entries are displayed.

Parameter	Description	
		When PIA is omitted, information from the JESplex tracking table is displayed instead. These are jobs that are active on all systems in the JESplex, or that are in the JES input queues ready to run.
	JOb=*jobname*	Limits the display to only those jobs matching the specified jobname. You can specify generic selection criteria (see "Global Event Selection Parameters" on page 262).
	TARget=*mvsname*	Limits the display to only those jobs active on the specified system.
STATus	Default. Displays a list of MVS systems and their Zeke-related status.	
SUMmary	Displays the status of Zekeplex services for this Zeke. Each line displays the status of a Zekeplex service.	
COUnts	Displays statistical information for the LPAR from which the command is issued.	
	CLEAR	Resets the statistics to zero.
START	Starts the specified Zekeplex service for this Zeke. Typically, you use this parameter to restart a Zekeplex service after an interruption has occurred. These are the valid keywords:	
	Notduring	Starts the XCF services for processing NOTDURING dependencies.
	Comm	Starts the XCF services for processing COMM records.
STOP	Stops the specified Zekeplex service for this Zeke. These are the valid keywords:	
	Notduring	Stops the XCF services for processing NOTDURING dependencies.
	Comm	Stops the XCF services for processing COMM records.

Parameter	Description

Note:

To be able to issue XCF start and stop requests, you must have `PLEXNOTD=YES` (for NOTDURINGs) and either `PLEXCOMM=YES` or `PLEXCOMM=XCFONLY` (for COMM records) set in your Zeke*xx* (PARMLIB) options member. See the *ASG-Zeke Scheduling for z/OS Installation Guide* for more information.

Command output is not displayed for start and stop requests. Refer either to your JES log or to the Zeke job log for the status of start and stop requests.

`PURGE` Removes an entry from the NOTDURING jobs table. This command can be used to remove a job (that no longer is active in the sysplex) if an error occurs during NOTDURING processing.

Caution! Because it does not correct any underlying errors, use this command only when necessary. Contact ASG Customer Support for assistance in determining the cause of the NOTDURING processing error.

These are the valid parameters:

`JOB=`*`jobname`*	Specifies the jobname.
`JOBID=`*`jesjobID`*	Specifies the JES job ID.
`TARGET=`*`mvsname`*	Optional. Specifies the z/OS system name where the job was running. (Omit this parameter to specify a job in the input queue.)

Note:

You can issue the ZPLEX DISPLAY command to identify the affected job and system.

`RELOAD` Invokes communications recovery and reloads the schedule. This action ensures that all memory tables are updated, with the exception of the resource table. (To manage the state of a resource, use the ZRESOURCE command.)

Note:

You can also use the SENDCOMM option of the ZALTER command to manually broadcast SQR refresh requests across the Zekeplex.

`COMM`	Required.

Sample Output—DISPLAY

This sample output is a result of issuing the command ZPLEX DISPLAY:

```
Z09B15I OS   JOBNAME  JOB ID   TYPE PH FL
Z09B15I SYSD D8NQ1RLM STC08510  STC 0E 80
Z09B15I SYSD JSMZSET  JOB01560 BTCH OD 83 HELD ASSIGNED
```

These are the column headings:

Heading	Description	
OS	Name of the system on which the job is currently active.	
JOBNAME	Job name.	
JOB ID	JES job ID.	
TYPE	Type of job. These are the possible types:	
	BTCH	Batch job.
	STC	Started task.
PH	Code used by ASG Customer Support.	
FL	Code used by ASG Customer Support.	
(no heading)	HELD	Indicates that the job is in the job queue, but is on hold.
	ASSIGNED	Indicates that the job is associated with an SQR that was dynamically-created by Zeke.

Sample Output—DISPLAY PIA

This sample output is a result of issuing the command ZPLEX DISPLAY:

```
Z09B16I JOBNAME   JOBID     ASID  STARTED   ACT
Z09B16I           JOB09935  003A  12:23:12   -
Z09B16I           JOB09897  003B  11:41:31   -
Z09B16I OASIS30A  STC09915  00FC  11:53:45   Y
Z09B16I ZEBB350S  STC09932  00BD  12:18:47   Y
```

These are the column headings:

Heading	Description
JOBNAME	Job name.

Heading	Description
JOBID	JES job ID.
ASID	Hexadecimal address space ID.
STARTED	Time that Zeke recorded the job in this table.
ACT	Table entry status. **Note:** This column is displayed only when the ALL keyword is coded. These are the possible statuses: - A hyphen (-) indicates the entry is available for reuse. Y Indicates the entry is active and will be used for scheduling decisions.

Sample Output—STATUS

This sample output is a result of issuing the command ZPLEX STATUS:

```
Z09B11I z/OS     NOTD ASG-Zeke COMM
Z09B11I ASGM113A XCF *MYZEKE1  DB
Z09B11I *unknwn* ---- MYZEKE2  DB
Z09B11I ASGM112A POLL *unknwn* ----
```

These are the column headings:

Heading	Description
z/OS	System name, as reported by XCF. If the system associated with the Zeke job cannot be determined (that is, if Zeke is not using XCF services), UNKNWN is displayed.
NOTD	Method being used for NOTDURING processing. XCF XCF messaging services are being used to acquire data from this system. POLL Enhanced Status subsystem calls are being used.
ASG-Zeke	CPU ID. An asterisk precedes the name of the Zeke that is processing the ZPLEX command. If Zeke is not active on the system, UNKNWN is displayed.
COMM	Method being used for COMM record processing.

Heading	Description	
	XCF	COMM records are transmitted via XCF processing.
	DB	COMM records are transmitted via the Zeke database.
	XCFO	Indicates that the Zeke system is running with PLEXCOMM=XCFONLY.

Sample Output—SUMMARY

This sample output is a result of issuing the command ZPLEX SUMMARY when XCF is in use.

```
Z09B13I Service   Status         ----- Counts ------ -Max-
Z09B13I NOTDURING Active         In =   0  Out=1490    80
Z09B13I COMM      Active         In =  48K Out=  60K   40
Z09B13I COMM-SQR  Active         In = 132K Out= 600K    4K
Z09B18I                          Dup=  20K Big=   0
```

Heading	Description	
Service	Zekeplex service name. These are the possible services:	
	NOTDURING	NOTDURING XCF use. NOTDURING messages notify Zeke of job start/stop events. The maximum message size is 80.
	COMM	Non-SQR communication record use. COMM messages notify Zeke to update in-memory tables. The maximum message size is 40.
	COMM-SQR	SQR updates sent between Zekes without requiring database I/O. COMM-SQR messages notify Zeke to update in-memory tables, and report the number of full SQR COMM records that have been sent or received via XCF. The maximum message size is 4096. (The actual message size is normally 1K to 2K.)
Status	Zekeplex service status. These are the possible statuses:	
	Active	Service is active and is using XCF services.
	Active, NOXCF	Service is active, but is not using XCF services.
	Not active	Service is not active.
Counts	Scaled number of XCF messages sent (OUT=) and received (IN=) by the service. If XCF is not in use, this field is left blank.	

Heading	Description
Max	Maximum XCF message size that has been sent or received for the service since the Zeke address space was last started. It is scaled in the same way as the Counts values. If you are using a Coupling Facility Resource Management policy, this value may be used to define the maximum message size.
Dup	Number of duplicate SQR messages that have been ignored. A duplicate message occurs when an event (such as a short-running job) is updated several times in quick succession. Only the most recent message is processed. This is informational only; no action is necessary.
Big	Number of SQR messages that are too large to be sent using the XCF messaging service (the XCF maximum size is 61K). Standard COMM processing is performed for these messages. If the number of Big messages is nonzero, examine the Zeke job log for Z013EW messages.

Sample Output—COUNTS

This sample output is a result of issuing the command ZPLEX COUNTS:

```
ZPLEX COUNT
Z09B20I                     IN        OUT        DUP       SIZE
Z09B20I NOTDURING            0          0                    0
Z09B20I SQR                  0          0          0         0
Z09B20I EVENT                0          0
Z09B20I RESOURCE             0          0          0
Z09B20I DQT WAITS            0         21
Z09B20I OTHER                0          4          0        24
Z09B20I VIA 05F                         0
Z09B20I SQR OVERFLOW         0
Z09B20I RES REVERSED         0
Z09B20I 05F1 BACKLOG         0
Z09B20I 05F2 BACKLOG         0
Z09B20I WEAK BACKLOG         0
Z09B20I DQT  BACKLOG        10  AT 2020/01/01 09:25:00
```

These are the column headings:

Heading	Description	
(no heading)	NOTDURING	NOTDURING XCF use. Communication records that notify Zeke of job start/stop events. The maximum message size is 80.
	SQR	Communication records that contain the full SQR.
	EVENT	Communication records that contain only the event number.

Heading	Description	
	RESOURCE	Communication records that contain a full resource.
	DQT WAITS	(Only if `PLEXCOMM=XCFONLY`) Communication records that contain wait reasons for events in the dispatch queue table.
	OTHER	Communication records that contain information other than full SQRs, event numbers, and DQT wait reasons.
	VIA 05F	Communications records queued to ZEKE05F rather than sent directly to XCF.
	SQR OVERFLOW	SQRs that are too large to be sent as a full SQR, and that were sent as just an event number instead.
	RES REVERSED	Number of resources acquired and immediately released because the resource table in storage was out of sync with the database.
	05F1 BACKLOG	Highest number of communications records remaining to be processed after the ZEKE05F1 subtask processed a batch of communication records. When not zero, it is followed by the date and time that the high watermark occurred.
	05F2 BACKLOG	(Only if `PLEXCOMM=XCFONLY`) Highest number of communications records that remain to be processed after the ZEKE05F2 subtask processed a batch of communication records. When not zero, it is followed by the date and time that the high watermark occurred.
	WEAK BACKLOG	Highest number of weak resolution requests queued to be processed by the weak resolution subtask. When not zero, it is followed by the date and time that the high watermark occurred.
	DQT BACKLOG	Highest number of entries that remain in the dispatch queue after the dispatcher completed a dispatch cycle. When not zero, it is followed by the date and time that the high watermark occurred.
IN	Number of messages received.	
OUT	Number of messages sent.	

Heading	Description
DUP	Number of duplicate messages that have been ignored. Only the most recent message is processed.
SIZE	Maximum message size that has been sent or received for the service since the Zeke address space was last started.

ZREFRESH Command

The ZREFRESH command refreshes an SQR by resetting the event as if it had not been run. The prerequisite and completion indicators are reset.

Because the REFRESH command resets all prerequisite completion indicators, all prerequisite conditions (e.g., EOJ or AEOJ.) must be satisfied again. To automatically satisfy the prerequisites, use the ZALTER WHENOK command.

Zeke automatically places a refreshed event on operator hold. The operator must release the hold before the event can be dispatched.

Only events that have been dispatched (events in Success, Active, or Pending status) can be refreshed using the ZREFRESH command.

Syntax



Parameters

To select an event(s), enter one or more of the global event selection parameters (starting on page 261). You can use these parameters with the ZREFRESH command:

Parameter	Description
RERUN	Adds the RERUN designation to the SQR. The RERUN designation appears in the ZDISPLAY output and is passed to the user exit ZEKE14D. If the Zeke generation option TrigRrn is set to N, the event will not trigger the WHEN conditions of other events. Use the NORERUN parameter of the ZALTER command to remove the RERUN designation. For example, this command sets up event 6 for processing again and adds the RERUN designation to the SQR: `ZREF EV 6 RERUN`
FORCE	Releases an event's resources before refreshing the event. This produces the same effect as releasing all of the resources and then issuing the ZREFRESH command. If there are no resources for the event, this parameter is ignored. For example, this command releases all resources and sets up event 12 for processing again: `ZREF EV 12 FORCE`

Examples

These examples show the use of some of the global event selection parameters.

This command sets up all job events that have an application ID of TST for processing again:

```
ZREFRESH JOB * APP TST
```

This command sets up all job events with a group ID of PRD for processing again:

```
ZREF JOB * GRO PRD
```

This example uses both ZREFRESH and ZALTER on the same command line. It sets up event 45 for processing again and consider all prerequisites satisfied:

```
ZREF EV 45 ZA EV 45 WHENOK
```

ZRELEASE Command

Use the ZRELEASE command to release these holds on Zeke processing:

- An operator hold on events
- An operator hold on an initiator
- An operator hold on a job class
- A system hold

See "ZHOLD Command" on page 361 for more information on the types of holds.

Releasing Events on Hold

Events could be held because of an operator hold or a system hold.

Operator Holds

To release an operator hold on one or more events, you use the global event selection parameters (see the listing beginning on page 261). The number of events selected for release are reported in the command output. An event can be selected for release only if it is on hold. Otherwise, this message appears:

```
Z0906W NO QUALIFYING SCHEDULE ENTRIES FOUND
```

System Holds

To override a system hold on one or more events, you also use the global event selection parameters (see the listing beginning on page 261). If any of the selected events are waiting in the dispatch queue because of a system hold, those events are released so that they can be dispatched (while the system hold remains in place).

Releasing a Job Class on Operator Hold

To release an operator hold on a job class, you specify the job class to be released. Releasing a hold on a job class enables Zeke to submit jobs in that class to JES.

Releasing an Initiator on Operator Hold

To release an operator hold on an initiator, you specify the initiator to be released. You can issue the command ZD AV to display the initiators currently on hold. . Releasing a hold on an initiator enables Zeke to dispatch job events to that initiator.

Releasing a Zeke System Hold

To release a system hold on Zeke, use the SYSTEM parameter. A system hold prevents Zeke from dispatching any events, except events released by a ZRELEASE command. A system release re-enables all Zeke event dispatching.

You must issue the ZRELEASE SYSTEM command from the Zeke system to be released. You cannot use the command to release another system.

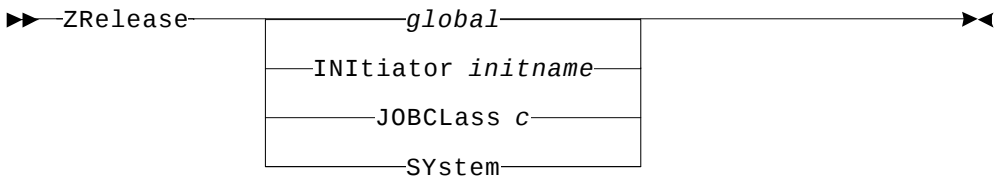
This command is accepted only if a system hold is currently in effect; otherwise, the command is rejected.

Releasing Remote Jobs on Hold

Through OASIS/DMS, when a remote system goes down and comes back up, Zeke automatically issues this command to release jobs held for the system while it was 'unregistered' from the OASIS/DMS network:

```
ZREL JOB * HOLD=NET TARGET=xxxx
```

Syntax



Parameters

You can use these parameters with the ZRELEASE command:

Parameter	Description

INItiator	Specifies the ID (up to two characters long) of the initiator to be released.
	Examples:
	This command releases the operator hold on initiator A:
	`ZR INI A`
	This command releases the operator hold on initiator 10:
	`ZR INI 10`

JOBCLass	Specifies the single-character job class to be released. For example: `ZR JOBCL A`

SYstem	Releases an existing hold on all event dispatching. For example, this command releases dispatching for the local Zeke system:
	`ZREL SYS`
	Note:
	You must issue this command from the Zeke system to be released. You cannot use the command to release another system.
	If you include a system ID with this command, the system ID is ignored. For example, if you issue this command from CPUB:
	`ZREL SYS CPUA`
	The result is that system CPUB is released instead of system CPUA.

Examples

These are some examples of using global event selection parameters with ZRELEASE.

This command releases the operator hold on event 45:

```
ZR EV 45
```

This command releases the operator hold on any job event having an eight-character jobname beginning with PR:

```
ZR JOB PR******
```

This command releases the operator hold on all events in the schedule queue or dispatch queue:

```
ZR ALL
```

This command releases the operator hold on all job events:

```
ZR JOB *
```

This command releases the operator hold on all job events that have been internally placed on NET hold:

```
ZR JOB * HOLD=NET
```

Note:

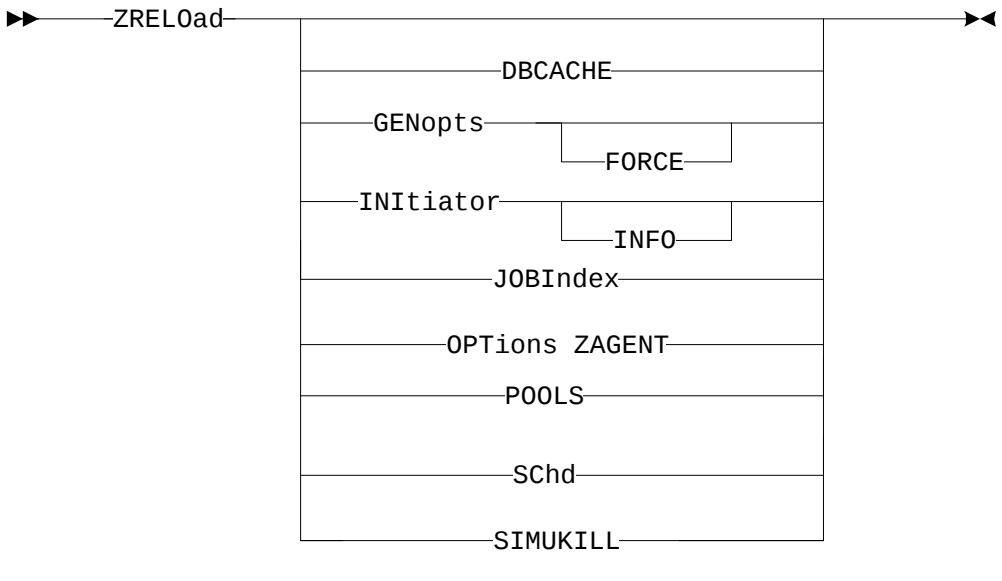
These are the valid qualifiers for HOLD=:

```
OPER
SJCL
NET
RFSH
DNPL
INTR
PSID
```

ZRELOAD Command

The ZRELOAD command reloads selected system tables. If you are running multiple systems, the ZRELOAD command must be issued on each system.

Syntax



Parameters

You can use these parameters with the ZRELOAD command:

Parameter	Description
DBCACHE	Reloads the database cache for this Zeke system.
GENopts	Reloads the system generation options. **Note:** You must exit from the Generation Options screen for the ZRELOAD GENOPTS command to take effect. See Chapter 7, "Generation Options," on page 519 for a list of generation options that cannot be reset by this command (i.e., where Zeke must be re-cycled before they are effective).

Parameter	Description
FORCE	Reloads the generation options without validating the GENOPT name. Normally, Zeke validates the GENOPT name when the ZRELOAD GENOPTS command is issued and rejects the request in these situations: • The default GENOPT ******** is the currently active GENOPT, and a new GENOPT has been added with a name that matches the current system name. • The name of the currently active GENOPT matches the system name, and that GENOPT has been deleted. The ZRELOAD GENOPTS command is rejected to prevent loading the wrong generation options.
INITiator	Reloads the system initiator tables and job class capacity limits.
INFO	After a successful reload, the ZRELOAD INITIATOR command also reports the new initiator definitions (by essentially issuing the ZD AVAIL INFO command (see "Displaying Initiator Information" on page 336).
JOBIndex	Reloads the job index in storage. If the DSPIndex generation option is set to Y, the index resides in a data space named ZIX (owned by the OASIS address space). If the EDBindex generation option is set to Y, the index resides in common storage. If neither option is set, this parameter is not valid with the ZRELOAD command.
OPTions	(Currently supported for the ZAGENT*xx* options member only.) Reloads and processes the specified Zeke options member. This is the valid option: ZAGENT Required. Reloads the ZAGENT*xx* options member.
POOLS	Reloads the active (in-memory) system pools from the Zeke database. Zeke reviews the schedule queue table (SQT) to determine whether any scheduled events are eligible for dispatch. In a Zekeplex, issuing this command on one Zeke system causes all other Zeke systems to reload their system pools and review their schedule tables.Reloads the active (in-memory) system pools from the Zeke database. Zeke reviews the schedule queue table (SQT) to determine whether any scheduled events are eligible for dispatch. In a Zekeplex, issuing this command on one Zeke system causes all other Zeke systems to reload their system pools and review their schedule tables.
SChd	Reloads the schedule tables for this Zeke system.

Parameter	Description
	Note: If the SCHTABLE=DSPACE start-up option is specified in the ZEKE*xx* options member, the schedule table index resides in a data space named ZTBLDASP (created by the OASIS started task and managed by the OASIS address space). If the SCHTABLE=DSPACE option is specified, the schedule tables reside in common storage.
SIMULKILL	Manually terminate the simultaneous schedule load process occurring in a Zekeplex (where multiple Zekes share a database). For example, if the SCHEDULE function terminates abnormally, Simuload processes continue to stand by for additional communication records until the SCHEDULE function is restarted. In this case, you could terminate the schedule table load process manually for each Zeke system.

ZRESOURCE Command

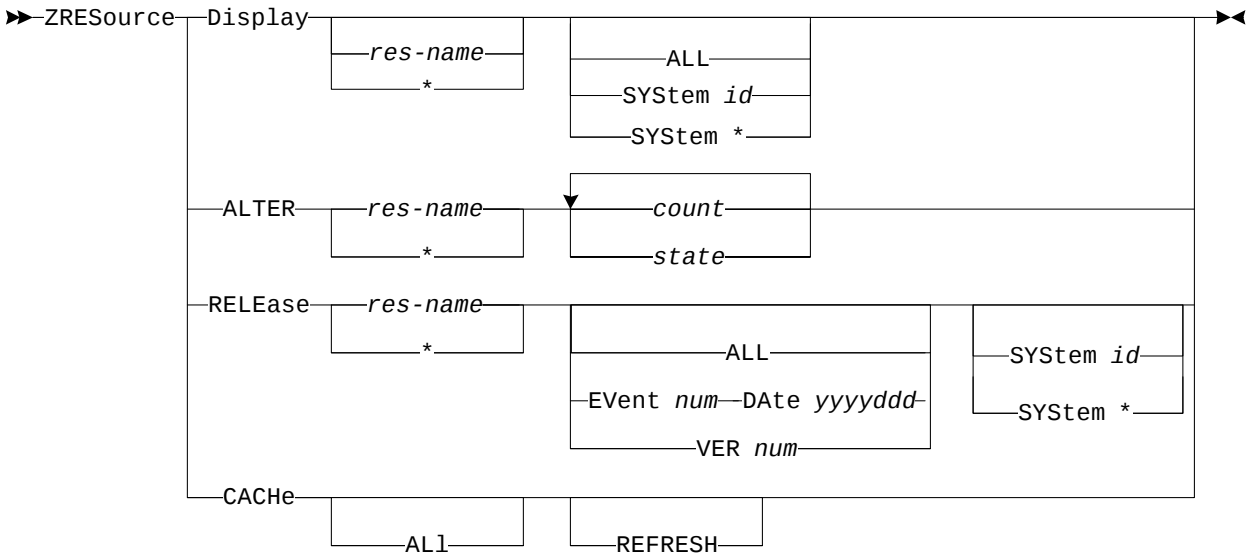
You use the ZRESOURCE command for these purposes:

- Maintaining the resources that are defined in the Zeke database. The ZRESOURCE command enables you to display or update resource details, and release a resource from an event(s) or system.
- Analyzing the entries in the memory cache that is used for Zeke resources and reporting any that are out of date. Stale entries may optionally be refreshed.

Note:

The cache is reported or updated only for the Zeke system where the ZRESOURCE command is issued.

Syntax



Parameters

You can use these parameters with the ZRESOURCE command:

Parameter	Description	
ALTER	Alters the resource count or state. These are the valid keywords:	
	count	Indicates how many events can use this resource at one time. These are the valid keywords:

Parameter	Description		
		`ALLSHR`	Allows all events to share this resource. For example: `ZRES ALTER 3840 ALLSHR`
		`NOSHR`	Prevents this resource from being shared by other events. For example: `ZRES ALTER 3840 NOSHR`
		`SHR` *n*	Indicates the number of events that can use this resource at one time. For example: `ZRES ALTER 3840 SHR 5`
	state		Indicates how many events can use this resource at one time. These are the valid keywords:
		`ENable`	Enables the resource for all events defined to use it. For example: `ZRES ALTER 3840 ENABLE`
		`DISABle`	Disables the resource for all events defined to use it. For example: `ZRES ALTER 3840 DISABLE`
`CACHe`	Requests a display of the Zeke resources cache. If you are experiencing problems with Zeke resources, you can issue the ZRESOURCE CACHE command on each Zeke system to determine whether the cause of the problem could be a stale cache. If the command reports out-of-date entries, first follow any actions as directed through message Z09DKI; then use the REFRESH keyword to refresh the cache with the current information from the Zeke database. **Caution!** Because the ZRESOURCE CACHE command reads every resource from the Zeke database and matches each one against the cache, this processing could take a long time if your site has many resources defined in the database.		
	`ALl`		Reports all entries (whether or not they are stale).

Parameter	Description	
	REFRESH	Requests that out-of-date entries be refreshed. When the REFRESH keyword is present, the CACHE request is treated as ZRESOURCE ALTER. **Note:** For Zeke command access checking, a CACHE request without the refresh keyword is treated as ZRESOURCE DISPLAY.
Display	Displays the resources. These are the valid keywords:	

	ALL	Displays additional information about enqueued logical resources. For example:
		`ZRESO DISPLAY * SYS * ALL`
		`Z0994I RESOURCE DISPLAY`
		`SHARE MAX EX SYSTEM RESOURCE NAME`
		`0 1 N ASGSYSB RESSYSB`
		`0 1 N (GLOBAL) RESSYSD`
		`0 1 N (GLOBAL) RESO3`
		`0 7 Y (GLOBAL) RESO2`
		`EXCL(6/0/2020134 A)`
		`5 40 N (GLOBAL) RESO1`
		`, SHR(5/0/2020131 A)`
		`Z0929I ZEKE COMMAND PROCESSED`
		See the *ASG-Zeke Scheduling Messages and Codes Guide* for details about the output.
		See the Resources chapter in the *ASG-Zeke Scheduling for z/OS User's Guide* for more information.

	SYStem *id*	Shows resource detail for all systems (*), the specified system ID, or the system from which you issue the ZRESOURCE command (default). For example, this command displays resource detail for the current system:
		`ZRES DISPLAY 3480`
		`ZRES DISPLAY 3480 SYS MVSA`
		Displays resource detail for the MVSA system.

RELEase	Releases the acquired or kept resource from one event or from all events. Either EVTNUM or ALL is required. These are the valid keywords:	
	ALL	Releases the resource from all events. For example: `ZRES RELE 3480 ALL`

Parameter	Description	
	EVtnum *num*	Releases the resource from the specified event. For example: `ZRES RELE 3480 EV 100`
	DAte *yyyyddd*	Required if you include EVTNUM. Specifies the SQR date (in *yyyyddd* format). The resource is released for this SQR only. For example: `ZRES RELE 3480 EV 100 DA 2020214`
	SYStem *id*	Specifies the system from which to release the resource. Defaults to the system from which the command is issued. You can enter * for all systems. For example: `ZRES RELE 3480 ALL SYS *`
	VER *num*	Releases the resource from the specified version of the event. If a version number is not specified, the resources are released for all versions of the event. For example: `ZRES RELE 3480 EV 100 VER 3`

res-name	Required. Resource names cannot contain embedded blanks. Enter an asterisk (*) to specify *all* resources.
	Examples:
	This command displays resource detail for resource 3480:
	`ZRES DISPLAY 3480`
	This command displays resource detail for all resources:
	`ZRES DISPLAY *`	

Sample Output

This is a sample of the output:

```
Z0994I RESOURCE DISPLAY
SHARE   MAX  EX SYSTEM   RESOURCE NAME
    0   ALL  N  (GLOBAL) RESOURCE1
    0   110  N  (GLOBAL) RESOURCE2
    0   222  N  (GLOBAL) RESOURCE3
    0   100  N  (GLOBAL) SHARED.RESOURCE
    0   100  N  (GLOBAL) FRED
    0   200  N  (GLOBAL) EXCLRESO
```

ZSCAN Command

The ZSCAN command validates or retrieves JCL that is submitted by Zeke. For example, for a z/OS job, the JCL is passed to z/OS with the TYPRUN=SCAN parameter specified on the job card.

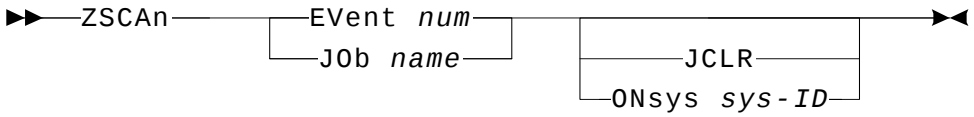
ZSCAN can be used for any job event in the schedule queue that is submitted to an operating system that supports this type of feature. The event's SQR is not affected in any way and the event is dispatched normally when all its dispatch requirements are met.

The default is to scan the event on the submitting system. If another system is desired, use the ONSYS parameter.

Note:

This command is not supported for SQRs which have JESQ as the JCL source.

Syntax



Parameters

Use the EVENT or JOB global selection parameter (descriptions starting on page 261) with the ZSCAN command. These are the valid parameters that you can use with the ZSCAN command:

Parameter	Description
JCLR	Retrieves the JCL from the JCL source and puts it in the SQR so you can view or update it. The JCL must reside on the same system you are issuing the command from. For example, this command retrieves the JCL for event 14: `ZSCAN EV 14 JCLR` **Note:** The JCLR parameter cannot be used with the ONSYS parameter.

Parameter	Description
ONsys	Scans the JCL on the specified system. For example, this command submits job PAY for scanning on system B (system B must match the platform type of the job): `ZSCAN JOB PAY ONSYS B` **Note:** The ONSYS parameter cannot be used with the JCLR parameter.

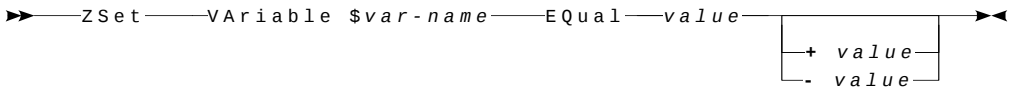
ZSET Command

The ZSET command sets a value.

Setting Variable Values

The ZSET command can be used to set the value of a Zeke variable to a specified or calculated value. The variable is added to the Zeke database if it does not already exist. The ZSET command cannot be used to set OASIS variable values.

Syntax



All Zeke variables must begin with a dollar sign ($).

If the variable value contains special characters, such as an asterisk (*), period (.), or ampersand (&), enclose the value within single quotes ('). For example, '***' or '='. Single quotes are not required for numeric values, unless the first digit is zero. If you do not enclose a number beginning with zero in single quotes, the leading zero is dropped. If you enclose a numeric variable value in single quotes, you cannot add to or subtract from the value using the plus and minus parameters. Also, if you enclose a date value in single quotes, you cannot perform date calculations on the variable.

If quote marks are part of the character string, enclose the string within another set of quotes. For example, if the string is 'KEEP THE QUOTES', enter ''KEEP THE QUOTES'' as the value.

All valid Zeke special names, such as ZEKECPU, DATEJ, TIME, and DATE, can be specified as the value of a Zeke variable. The variable value becomes the value represented by the Zeke special name.

Examples

This command sets $CHAR3 is equal to the current date:

```
ZSET VAR $CHAR3 EQ DATE
```

This command sets $CHAR4 is equal to the CPU name:

```
ZSET VAR $CHAR4 EQ ZEKECPU
```

Parameters

You can use these parameters with the ZSET command:

Parameter	Description

EQual	Precedes the new value for the variable.
	Examples:
	This command sets $TEST equal to YES (character value):
	`ZSET VAR $TEST EQ YES`
	This command sets $TEST1 equal to 456 (numeric value):
	`ZSET VAR $TEST1 EQ 456`
	This command sets $TEST2 equal to $TEST1:
	`ZSET VAR $TEST2 EQ $TEST1`
	This command sets $CHAR1 is equal to 'A CHAR STRING' (character value):
	`ZSET VAR $CHAR1 EQ 'A CHAR STRING'`

VAriable	Specifies the name of the variable to be set.

Note:

You can use both a plus (+) and a minus (-) on the same statement; however, only one plus and one minus can be used per statement.

For example, this command is not valid:

```
ZSET VAR $VAR1 EQ $A + $B - $C is valid, but ZSET VAR $VAR1 EQ $A + $B + $C
```

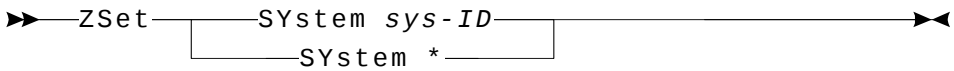
+ (plus)	Adds the specified value to the value following EQ. You can specify to add a numeric value or a variable. For example, this command increments variable $TEST3 by one: `ZSET VAR $TEST3 EQ $TEST3 + 1`
- (minus)	Subtracts the specified value from the value following EQ. You can specify a numeric value or a variable. For example, this command increases $TEST3 by 4, plus the value of $TEST1, minus the value of $TEST2: `ZSET VAR $TEST3 EQ 4 + $TEST1 - $TEST2`

See the *ASG-Zeke Scheduling for z/OS User's Guide* for more information on setting Zeke variable values, and using variables to trigger events, and using variables to restart a job.

Setting the Default System ID

The ZSET command issued with the SYSTEM parameter changes the default system ID for subsequent operator commands. This only affects the commands entered from the user that enters the ZSET command. The system ID is not affected.

Syntax



Normally, the Zeke command processor assumes the commands entered are from the system from which the command originates. To enter commands for events on another system ID, you must enter the system ID on the command line.

This command is useful for CMS users, since each CMS user accessing the same Zeke database must have a unique system ID. Events displayed in the ZCOM function are always on a different system ID (that of an OS user ID). Use this command to establish a default system ID for subsequent commands (even the operator console command processor on System A could be set to affect events on System B by default). The ZSET command with the SYSTEM parameter is allowed even if other types of ZSET commands are restricted through Zeke's security system.

Examples

This command sets the default system to system B. All operator commands are assumed to apply to system B until the system is reset:

```
ZSET SYSTEM B
```

This command displays events on system B and is the same as entering `ZD SYS B`:

```
ZD
```

This command changes event 4 from system A to system B:

```
ZA EV 4 NEWSYS A
```

This command displays events on system C:

```
ZD SYS C
```

This command causes subsequent Zeke commands to effect qualified events on ALL systems:

```
ZSET SYS *
```

ZSTATUS Command

The ZSTATUS command displays the reason an event is waiting in the dispatch queue. The output from the command is displayed on the system console. (If the event is simply waiting for an initiator, the reason is not displayed.) Some of the reasons include:

- Waiting for a required number of available tape drives
- Waiting for an operator OK

See Event Status/Reason Code on page 443 for descriptions of all the reason codes.

Any message displayed at the time operator action was first required is redisplayed.

The ZSTATUS command displays the same information as the ZD WAIT command on page 329.

Only events in the dispatch queue are selected for processing by the ZSTATUS command.

See the *ASG-Zeke Scheduling for z/OS User's Guide* for information on using Schedule View to monitor events.

Syntax



Parameters

The Global Event Selection parameter DQ (dispatch queue) is assumed for this command. The parameter SQ (schedule queue) is ignored if entered. Use any other global event selection parameters (starting on page 261) to select the proper events.

Examples

These examples illustrate the use of the ZSTATUS command (along with some of the global event selection parameters).

This command displays status of event 88:

```
ZSTATUS EV 88
```

This command displays status of job CD07UPDT:

```
ZST JOB CD07UPDT
```

Zeke Address Space Commands

Zeke provides its own operator modify commands for altering the Zeke environment. The Zeke main program (ZEKE47A), which manages the Zeke environment, is called by SSS4001 (the main program module for the Zeke started task). The Zeke started task creates the subtasks that execute in the address space. Each subtask supports a key Zeke feature and runs independently of other subtasks. Generally, the Zeke main task manages all subtasks, but, occasionally, you might need to intervene manually.

Note:

When the Zeke started task starts the Zeke server subtask (ZEKE6SRV), only the subsystem name is included as a parameter.

By default, Language Environment (LE) runtime options are set for a default LE environment. You can update the runtime options in memory for the Zeke server subtask by issuing a modify command to the Zeke address space. The command prefix is #. These are the valid command formats:

```
F xxxZ620A,#command

MODIFY xxxZ620A,#command
```

The address space commands described in this section are operator commands you can use to change the in-memory LE runtime options for a subtask. These changes persist only until the address space is restarted. Address space commands do not change the runtime options for an active subtask. For the changes to take effect, the subtask must be restarted. Address space commands can be issued against all subtasks, unless noted otherwise.

See Appendix C, “Other Address Space Commands,” on page 629 for additional commands that are supported by the ZEKE6SRV module only.

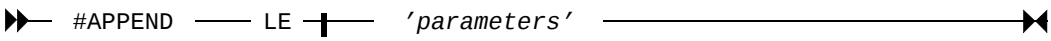
Note:

Zeke address space commands (including ZKILL) are authenticated only through external security (if enabled). By default, commands are authenticated against profiles created in the Z$CMD class in the external security product. See *ASG-Zeke Scheduling for z/OS User’s Guide* for more information on command security.

#APPEND

Use the #APPEND command to append LE parameters to the list of LE parameters used when the Zeke server subtask starts. The Zeke server must be restarted for the updated parameters to take effect.

Syntax



Parameters

You can use these parameters with the #APPEND command:

Parameter	Description
LE	Append LE runtime parameters.
'parameters'	Enclose parameters in single quotes. For example: `F xxxZ620A,#APPEND LE 'RPTOPS(ON),RPTSTG(ON),HEAPCHK(ON,1,0,0)'` Refer to your IBM *Language Environment Programming Reference* for valid LE parameters.

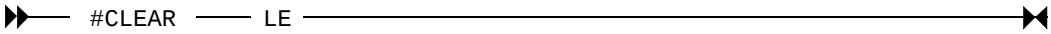
Sample Output

```
Z470LI #APPEND LE 'RPTOPTS(ON),RPTSTG(ON),HEAPCHK(ON,1,0,0)'
Z470SI NEW:RPTOPTS(ON),RPTSTG(ON),HEAPCHK(ON,1,0,0)
```

#CLEAR

Use the #CLEAR command to remove any LE parameters from the internal LE buffer that is passed to the Zeke server at attach time.

Syntax



Parameter

This parameter is required with the #CLEAR command:

Parameter	Description
LE	Clear LE runtime parameters. For example: F *xxx*Z620A,#CLEAR LE

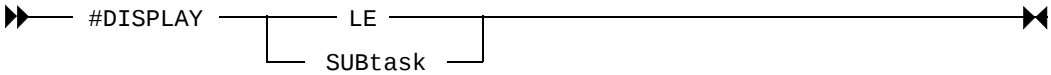
Sample Output

```
Z470LI #CLEAR LE
Z470SI NEW:
```

#DISPLAY

Use the #DISPLAY command to display the contents of the LE parameter buffer and subtask information for all subtask modules.

Syntax



Parameters

You can use these parameters with the #DISPLAY command:

Parameter	Description
`LE`	Display LE runtime parameters, including those currently active and those that are used the next time the Zeke server is started. For example: `F xxxZ620A,#DISPLAY LE`
`SUBtask`	Display subtask information for all subtask modules managed by ZEKE47A. For example: `F xxxZ620A,#DISPLAY SUB`

Sample Output

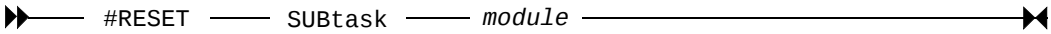
```
Z470LI #DISPLAY LE
Z470SI ATT:RPTOPTS(ON),RPTSTG(ON),HEAPCHK(ON,1,0,0)
Z470SI NEW:RPTOPTS(ON),RPTSTG(ON),HEAPCHK(ON,1,0,0)
```

```
Z470LI #DISPLAY SUBTASK
Z470KI Zeke subtask ZEKE45J  State RUNNING  Abends 000 Max 020  007F8680
Z470KI Zeke subtask ZEKE49A  State RUNNING  Abends 000 Max 020  007DF430
Z470KI Zeke subtask ZEKE45AT State RUNNING  Abends 000 Max 020  007DF980
Z470KI Zeke subtask ZEKE45AD State RUNNING  Abends 000 Max 020  007DF6D8
Z470KI Zeke subtask ZEKE05F1 State RUNNING  Abends 000 Max 020  007DFBB8
Z470KI Zeke subtask ZEKE05F2 State STOPPED  Abends 000 Max 020
Z470KI Zeke subtask ZEKE05H  State RUNNING  Abends 000 Max 020  007DEBE0
Z470KI Zeke subtask ZEKE05I  State RUNNING  Abends 000 Max 020  007DE938
Z470KI Zeke subtask ZEKE05G  State RUNNING  Abends 000 Max 020  007DEE88
Z470KI Zeke subtask ZEKE03T  State RUNNING  Abends 000 Max 020  007DF188
Z470KI Zeke subtask ZEKE6SRV State RUNNING  Abends 000 Max 020  007DE0C0
Z470KI Zeke subtask ZEKE48A1 State RUNNING  Abends 000 Max 020  007C8D90
```

#RESET

Use the #RESET command to reset the subtask restart counter back to zero. The subtask restart counter determines how many times a subtask has been restarted via automatic subtask recovery. This value is compared with the subtask maximum restart value to determine whether an automatic subtask restart should be performed.

Syntax



Parameters

You can use these parameters with the #RESET command:

Parameter	Description
SUBtask	Reset the restart counter back to zero for the specified subtask.
module	Subtask for which to reset the restart counter. For example: F *xxx*Z620A,#RESET SUB ZEKE6SRV

Sample Output

```
Z470LI #RESET SUB ZEKE6SRV
Z470KI Zeke subtask ZEKE6SRV State RUNNING Abends 000 Max 020
```

#SET

Use the #SET command to set these parameters or values:

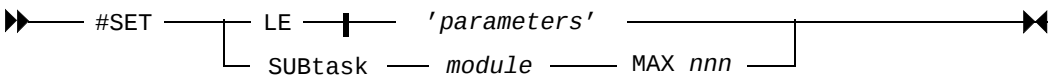
- LE parameters to be used when the Zeke server is attached.

  Note:

  In contrast to the #APPEND command, the #SET command clears the current parameters before adding the new ones.

- Maximum number of automatic restarts to be performed on a subtask that terminates unexpectedly.

Syntax



Parameters

You can use these parameters with the #SET command:

Parameter	Description
LE	Set the LE runtime parameters used when the Zeke server is attached. For example: `F xxxZ620A,#SET LE` **Note:** Using this parameter alone clears the existing parameters.
'parameters'	Enclose parameters in single quotes. For example: `F xxxZ620A,#SET LE 'RPTOPS(ON),RPTSSTG(ON)'` Refer to your IBM *Language Environment Programming Reference* for valid LE parameters.
SUBtask	Set the module restart count maximum for the specified subtask. For example: `F xxxZ620A,#SET SUB ZEKE6SRV MAX 40`
module	Subtask for which to set LE parameters.

Parameter	Description
MAX *nnn*	Maximum number of automatic restarts that are performed for a subtask that terminates unexpectedly. The valid values range from 1 through 999. For example: F *xxx*Z620A,#SET SUB ZEKE6SRV MAX 40

Sample Output

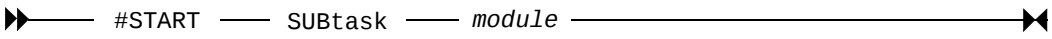
```
Z47OLI #SET LE 'RPTOPS(ON),RPTSSTG(ON)'
Z470SI NEW:RPTOPS(0N),RPTSTG(ON)
```

```
Z470LI #SET SUB ZEKE6SRV MAX 40
Z470SI Zeke subtask ZEKE6SRV State RUNNING Abends 000 Max 040
```

#START

Use the #START command to start a subtask module that is stopped. When a subtask is started, its abend counter is reset and the subtask is eligible for automatic restart.

Syntax



Parameters

You can use these parameters with the #START command:

Parameter	Description
SUBtask	Start the specified subtask.
module	Name of the subtask to be started. For example: F *xxx*Z620A,#START SUB ZEKE6SRV

Sample Output

```
Z470LI #START SUB ZEKE6SRV
Z470KI Zeke subtask ZEKE6SRV State RUNNING Abends 000 Max 020
Z6003I ZEKE OpsCentral Server enabled
```

Zeke Server Commands

The Zeke server executes as a subtask of the Zeke started task. One of its key functions is to facilitate communication between Zeke and a remote system (e.g., an OpsCentral client or a Zena server) by serving as the message processor. You issue commands to the Zeke server using a MODIFY command. The server command prefix is $. These are the valid command formats:

```
F xxxZ620A,$command

MODIFY xxxZ620A,$command
```

The server commands allow you to manage thread and trace options while the Zeke server is executing, and without having to restart it.

Command output is displayed on the operator console as WTO messages. (Responses to requests made using raw Zeke commands issued from OpsCentral are displayed on the OpsCentral console.)

Note:

Zeke server commands are authenticated only through external security (if enabled). By default, commands are authenticated against profiles created in the Z$CMD class in the external security product.

See the *ASG-Zeke Scheduling for z/OS Installation Guide* for more information on configuring the Zeke server and authenticating client sessions.

See Chapter 9, " Security," on page 423 for more information on command security.

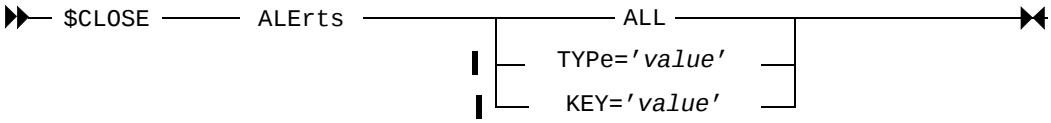
$CLOSE

Use the $CLOSE command to create a request to close an alert. The requested operation is placed on the queue for the alert processing thread that runs in the background for the Zeke server. You can close all alerts, individual alerts, or groups of alerts based on specified criteria (type or key value).

Note:

Use the $DISPLAY ALERTS command (see "$DISPLAY" on page 410) to display a list of all alerts in the cache (including type and key values.)

Syntax



Parameters

You can use these parameters with the $CLOSE command:

Parameter	Description
ALErts	Indicates alerts.

ALL	Close all alerts currently in the alert cache. For example, any of these commands closes all alerts:
	`F ZEKE620A,$CLOSE ALERTS ALL`
	`F ZEKE620A,$CLOSE ALL ALERTS`
	`F ZEKE620A,$CLOSE ALERTS TYPE="*" KEY="*"`
	`F ZEKE620A,$CLOSE ALERTS TYPE="*"`
	`F ZEKE620A,$CLOSE ALERTS KEY="*"`

KEY=	Close alerts with the specified key value.
	Note:
	You can use wildcard and placeholder characters in your selection criteria. See "Wildcard and Placeholder Characters" on page 259.
	Examples:
	This command closes alerts related to abends:
	`F ZEKE620A,$CLOSE ALERT KEY="*/AEO?"`
	This command closes alerts related to late jobs:
	`F ZEKE620A,$CLOSE ALERT KEY="*/Z0302I/*"`
	This command closes the alert with the specified key value:
	`F ZEKE620A,$CLOSE ALE KEY="plexid/Z0302I/000116/2020210/00001"`
	Either of these commands closes all alerts:
	`F ZEKE620A,$CLOSE ALERTS KEY="*"`
	`F ZEKE620A,$CLOSE ALERTS TYPE="*" KEY="*"`

Parameter	Description

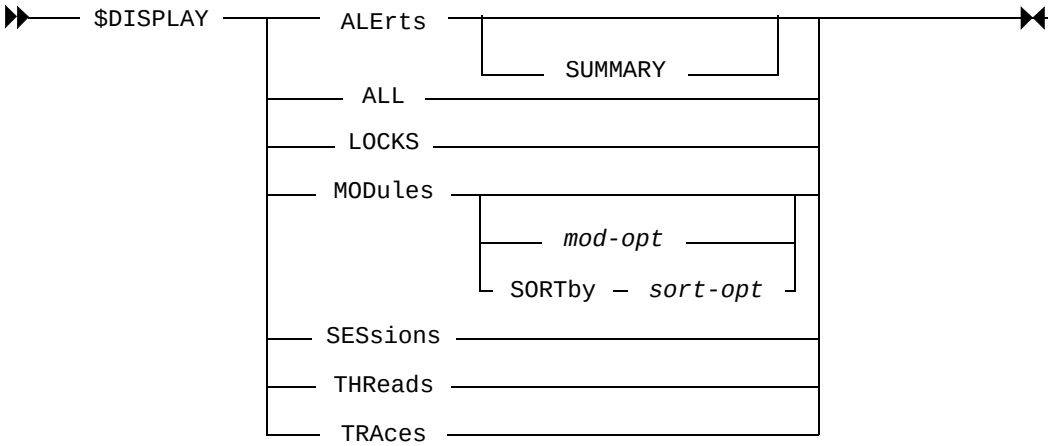
TYPe=	Close all alerts of the specified type.
	Note:
	You can use wildcard and placeholder characters in your selection criteria. See "Wildcard and Placeholder Characters" on page 259.
	Examples:
	This command closes alerts related to abends:
	`F ZEKE620A,$CLOSE ALERT TYPE="FAIL"`
	This command closes alerts related to late jobs:
	`F ZEKE620A,$CLOSE ALERT TYPE="LATE"`
	Either of these commands closes all alerts:
	`F ZEKE620A,$CLOSE ALERTS TYPE="*" KEY="*"`
	`F ZEKE620A,$CLOSE ALERTS TYPE="*"`

$DISPLAY

Use the $DISPLAY command to display these types of server and module information:

- Current sessions for the Zeke server
- Zeke server thread statistics (summary or detail)
- OpsCentral alerts
- Current trace settings for the Zeke server
- Current statistics for Zeke server locks
- Information about the modules currently loaded in storage

Syntax



Parameters

You can use these parameters with the $DISPLAY command:

Parameter	Description
ALErts	Display all alerts in the alert cache. F *xxx*Z620A,$DISPLAY ALErts
ALL	Display sessions, threads, traces, and alerts. F *xxx*Z620A,$DISPLAY ALL
LOCKS	Display the current statistics for Zeke server locks. Any thread holding or waiting for a lock is also displayed. The minimum, average, and maximum wait times are displayed for each lock. F *xxx*Z620A,$DISPLAY LOCKS
MODules	Display information about the modules currently loaded in storage (along with a private and common storage summary). Output will appear in the Zeke address space job log and the system log. F *xxx*Z620A,$DISPLAY MODules These are the valid module options: ALL — Default. Select modules from the Job Pack Area (JPA) and Common System Area (CSA) areas. CSA — Select modules from the CSA only. JPA — Select modules from the JPA only.

Parameter	Description	
	`LE`	Select modules from the Language Environment (LE) only.
`SORTby`	Sorting method for the displayed output. These are the valid sorting options:	
	`DATE`	Sort by compile/assemble date (which includes the time).
	`ENTRY`	Sort by entry point address.
	`LOAD`	Sort by load address.
	`MODULE`	Default. Sort by module name.
	`NONE`	Do not sort the information.
	`PTF`	Sort by PTF name.
	`SIZE`	Sort by module size.
	`SUBPOOL`	Sort by subpool number.
`SESsions`	Display the current sessions for the Zeke server. `F` *`xxx`*`Z620A,$DISPLAY SESsions`	
`THReads`	Display Zeke server thread statistics. Information for all thread classes and workers is displayed on the console. `F` *`xxx`*`Z620A,$DISPLAY THReads`	
`TRAces`	Display the current trace setting for the Zeke server. `F` *`xxx`*`Z620A,$DISPLAY TRAces`	

Sample Output

$DISPLAY ALERTS

```
$DISPLAY ALERTS
Z6092I $DISPLAY ALERTS
Z6109I 1 alerts found
Z6110I alert status=OPEN created=Thu Jan  4 10:07:07 2020
Z6111I    type=SYSHOLD key=plexid/Z0514W/SYSD
Z6112I    Z0514W Notice ** Zeke System is currently on Hold
```

$DISPLAY ALERTS SUMMARY

```
```

$DISPLAY LOCKS

```
$DISPLAY LOCKS
Z6126I Lock Name                                          Lock Count    Min      Avg      Max
Z6127I     Thread Ident.    --Date-- --Time-- -State-- Wait Sec Mode
Z6128I SCHED UPDATE                                                0 0.000000 0.000000 0.000000
Z6128I SCHED TABLE                                                 2 0.000068 0.000096 0.000125
Z6128I ALERT TABLE                                               200 0.000039 0.000489 0.020865
Z6129I     2557638000000005 10/16/17 16:49:02 *locked* 0.008700 Write
Z6128I JES UPDATE                                                  0 0.000000 0.000000 0.000000
Z6128I SESSION TABLE                                              29 0.000051 0.000117 0.000739
Z6128I SNMP GLOBAL                                                63 0.000044 0.004054 0.114910
```

$DISPLAY SESSIONS

```
Z6092I $DISPLAY SESSIONS
Z6098I Session list size 50 count 5
Z6099I Session Identifier                Userid   Date     Time     Client
Z6100I 0B71BB7347B481049A3E99070B2FDC4A DV001    10/13/17 15:43:35 testclient
Z6100I 324282145EFBDBA7C309087621BE2C1D DV001    10/13/17 15:43:30 testclient
Z6100I 60782A3FE221268D4DE7DC4D323653A0 DV001    10/13/17 15:43:32 testclient
Z6100I 69ABFD3D3DD6683567A5BF2762F678DB DV001    10/13/17 15:42:51 testclient
Z6100I 6D7B7326460B224E018460BFC097AEB0          10/13/17 15:43:32 testserver
```

$DISPLAY THREADS

```
Z6092I $DISPLAY THREADS
Z6086I Threads total=7 idle=6 busy=1 queued=0 complete=5
Z6087I Class=UNSOL min=1 max=1 total=1 idle=1 busy=0 queued=0 complete=2
Z6088I     Threadid         State Userid  Timestamp         Session
Z6089I     248B5A3000000004 idle          10/13/17 15:01:20
Z6087I Class=REQUEST min=5 max=5 total=5 idle=5 busy=0 queued=0 complete=0
Z6088I     Threadid         State Userid  Timestamp         Session
Z6089I     248C1C4000000005 idle          10/13/17 15:01:20
Z6089I     248C431000000006 idle          10/13/17 15:01:20
Z6089I     248C83C000000009 idle          10/13/17 15:01:43
Z6089I     248C9DA00000000A idle          10/13/17 15:01:43
Z6089I     248CF8300000000B idle          10/13/17 15:01:43
Z6087I Class=COMMAND min=1 max=1 total=1 idle=0 busy=1 queued=0 complete=3
Z6088I     Threadid         State Userid  Timestamp         Session
Z6089I     248C500000000007 busy  CONSOLE 10/13/17 15:01:20
```

Message	Label	Description	
Z6086I (displays totals for all thread classes)	Threads total	Total number of threads in all thread classes. These are the possible thread states:	
		`idle`	Total number of idle threads in all thread classes.
		`busy`	Total number of busy threads in all thread classes.
		`queued`	Total number of queued requests waiting for an idle thread in all thread classes.
		`complete`	Total number of completed requests in all thread classes.
Z6087I (displays totals for a specific thread class)	Class	Thread class name. These are the possible classes:	
		`REQUEST`	OpsCentral client requests.
		`COMMAND`	Console operator requests.
		`UNSSQR`	Unsolicited SQR requests.

Message	Label	Description	
		UNSALERT	Unsolicited alert requests. These are the alert types: • EARLYWNG (early warning) • RISREG (RI Server registration) • MONITOR (thread activity monitor) • HTTP (Web Services thread) • HTTPS (Web Services with Secure Sockets) • AGENTREQ (Zena protocol request) • AGENTUNS (Zena protocol unsolicited status messages)
		ZENACOM	Sends and receives messages for Zena agent connections.
		ZENADISP	Dispatches work from Zeke to a Zena agent (creates messages that ZENACOM sends).
		ZENACTL	Handles the connection to and initialization of a new Zena agent.
		ZENAREQ	Zena agent requests;.processes messages received from a Zena agent.
	min	Minimum number of threads for this thread class.	
	max	Maximum number of threads for this thread class.	
	total	Total number of threads in this thread class.	
	idle	Number of idle threads in this thread class.	
	busy	Number of busy threads in this thread class.	
	queued	Number of queued requests waiting for an idle thread in this thread class.	
	complete	Number of completed requests in this thread class.	
Z60088I/Z6089I	Threadid	Thread name, if the thread class has a name. Otherwise, the thread class number is displayed.	
	State	State of the thread. These are the possible thread states:	
		idle	Thread is waiting for work to do.

Message	Label	Description	
		`busy`	Thread is currently processing a request.
		`quies`	Thread is being terminated.
		`dead`	Thread has been terminated and will be deleted soon.
	User ID	User ID, if one is associated with the thread.	
	Timestamp	For idle threads, this is the start time. For busy threads, this is the time at which the thread's state changed from idle to busy.	
	Session	Session ID, if one is associated with the thread.	

$DISPLAY TRACES

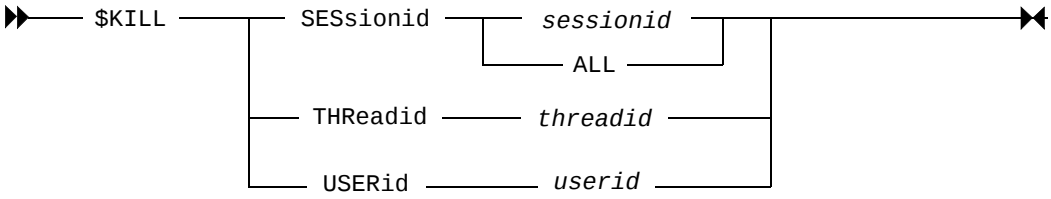
```
Z6092I $DISPLAY TRACES
Z6083I Trace messages issued:
Z6084I   function
Z6084I   /memory
Z6084I   thread
Z6084I   stats
Z6084I   fullstats
Z6084I   workflow
Z6084I   command
Z6084I   message
Z6084I   msgcontent
Z6084I   parser
```

$KILL

Use the $KILL command to perform these actions:

- Remove specific (or all) session IDs from the Zeke server's list of sessions.
- Terminate a thread that appears to be hung.
- Remove all sessions associated with a specified user ID.

Syntax



Parameters

You can use these parameters with the $KILL command:

Parameter	Description
SESsionid	Remove a session ID from the Zeke server's list of sessions. **Note:** To display the session list, issue the $DISPLAY SESSION command.
sessionid	32 hexadecimal session ID to remove from the session list. `F xxxZ620A,$KILL SES 8F272A2AD47EDE71A32CE263C5402414`
ALL	Remove all sessions from the session list.
THReadid	Terminate a thread that appears to be hung. If the thread is one that can be interrupted, then it is terminated. When a thread is terminated, it is removed from the thread class. **Note:** To display the thread IDs, issue the $DISPLAY THREAD command. Any threads that are already terminated will appear temporarily in the display with a 'dead' status, until they are removed from the thread class by the thread manager. **Caution!** If a thread is currently processing a $KILL command, it cannot be terminated. Any request to do so will fail. **Note:** If work is submitted to a thread class that has no available thread to process the work, the thread manager creates a new thread to process the work (as long as the maximum number of threads permitted for the class is not exceeded). If all threads in the class are busy, the work request waits until a thread is idle and can accept the new work request.
threadid	16-hexadecimal thread ID to be terminated. `F xxxZ620A,$KILL THR 248DAD5000000004`
USERid	Remove all sessions associated with a particular user ID.
userid	User ID for which to remove all sessions. `F xxxZ620A,$KILL USER DV001`

Sample Output

$KILL SESSIONID

```
Z6092I $KILL SES 8F272A2AD47EDE71A32CE263C5402414
Z6101I 8F272A2AD47EDE71A32CE263C5402414 session removed
```

$KILL USERID

```
Z6092I $KILL USER DV001
Z6101I 030B69F366571AD8DBA39A0B3E0AF27E session removed
Z6101I 3CADA2EEB7106341C3B8311CC35FDA80 session removed
Z6101I 62E365A9B63D78671F31AF9F1ACE3F57 session removed
Z6101I 8D9E769038CB56CE7B0ED2334C38FA9B session removed
Z6101I F9C331186777E23BAAC2A1CE63F6F855 session removed
Z6091I Sessions for userid DV001 have been removed
```

Terminating Zeke Server Threads After ZKILL

If a server thread is hung after ZKILL is issued, the Zeke server subtask might not terminate without operator intervention. If this situation develops, follow this procedure to terminate a thread manually.

To obtain a thread ID

1. Issue the `SDSF PS` command to display a list of threads for a process.
2. Enter `D` next to the line displaying the Zeke address space.
3. Locate the process ID (PID) and the thread ID.

To terminate a hung thread

- Issue this console command:

  `/F BPXOINIT,TERM=`*`process_id.thread_id`*

  For example, if you received this message:

  ```
  Z6136W Zeke Server Thread 262280.255E07F00000000A busy ...
  ```

  You would enter this console command to terminate the hung thread:

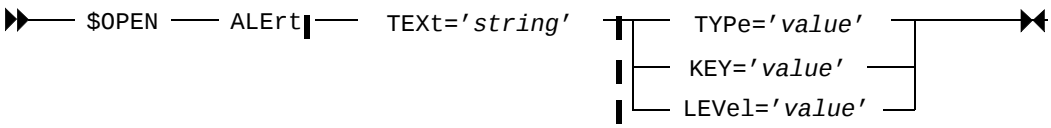
  ```
  /F BPXOINIT,TERM=262280.255E07F00000000A
  ```

After all threads have been terminated, the Zeke server can terminate normally and enable the Zeke address space to terminate normally.

$OPEN

Use the $OPEN command to create an alert from the operator console to be displayed in an OpsCentral client alert window.

Syntax



Parameters

You can use these parameters with the $OPEN command:

Parameter	Description
ALErt	Indicates a alert.
TEXt=	Alert text (up to 80 characters long) to be displayed in the OpsCentral client alert window..
TYPe=	Type of alert (up to 12 characters long) to create. The default type is CONSOLE. **Note:** You cannot specify an asterisk (*) or a question (?) mark in the type. These characters are reserved for use as wildcard and placeholder characters in the selection criteria for Zeke operator commands.
KEY=	Key value (up to 60 characters long) for the alert to be created. This is the valid format: *plexid/message_ID/session_ID* This is the default key: *plex_ID*/CONSOLE/*session_ID* where *session_ID* is a unique 32-character hexadecimal value. **Note:** You cannot specify an asterisk (*) or a question (?) mark in the key. These characters are used as wildcard and placeholder characters in the selection criteria for Zeke operator commands.
LEVEL=	Alert level. These are the valid values: E Error.

Parameter	Description	
	W	Warning.
	I	Default. Informational.

Examples

```
F ZEKE610620A,$OPEN ALERT TEXT='System coming down in 5 minutes'
TYPE='SHUTDOWN' LEVEL='WARNING' KEY='SYSD/SHUTDOWN'

F ZEKE610620A,$OPEN ALERT TEXT='No tape drives are available on
SYSD' TYPE='RESOURCE' KEY='SYSD/TAPES'

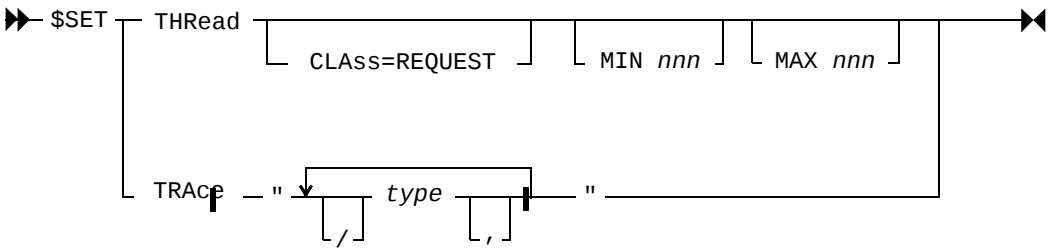
F ZEKE610620A,$OPEN ALERT TEXT='The phones are not working in
Houston, call Miami for assistance'
```

$SET

Use the $SET command to perform these actions:

- Set the number of threads used to process requests. As requests are received, they are given to a worker thread for processing. If there are not enough threads available, the request waits until one is available. You can set or change the number of worker threads.
- Set or reset the trace flags.

Syntax



Parameters

You can use these parameters with the $SET command:

Parameter	Description
THRead	Sets the number of threads used to process requests.

Parameter	Description	
CLAss	Thread class. These are the valid values:	
	name	Thread class name.
	REQUEST	Default. OpsCentral request class. The REQUEST class is used to process all request from OpsCentral clients.
MIN *nnn*	Minimum number (1 through 999) of threads to use for the specified class. For example: F *xxx*Z620A,$SET THREAD MIN 10	
MAX *nnn*	Maximum number (1 through 999) of threads to use for the specified class. For example: F *xxx*Z620A,$SET THREAD MAX 20 F *xxx*Z620A,$SET THREAD MIN 3 MAX 8	
TRAce	Sets (or resets) the specified trace flags.	
type	Type of trace messages to generate. Specify a value to enable the trace flag. Precede the value with a slash (/) to reset (i.e., disable) the specified trace flag. These are the valid values:	
	ALL	All trace messages. Examples: This command enables all trace messages: F *xxx*Z620A,$SET TRACE "ALL" This command disables all trace messages: F *xxx*Z620A,$SET TRACE "/ALL" When specifying the ALL parameter, you can include additional type values to indicate an exception. For example, this command enables all trace messages *except* heap allocation/free messages: F *xxx*Z620A,$SET TRACE "ALL /MEMORY"
	FUNCTION	Function call entry/exit trace messages. For example, this command enables function call trace messages: F *xxx*Z620A,$SET TRACE "FUNCTION"
	MEMORY	Heap allocation/free trace messages.
	THREAD	Thread creation, destruction, and status change trace messages.

Parameter	Description	
	STATS	Summary status information. This information is logged every minute and includes thread activity and table usage.
	FULLSTATS	Detailed usage message statistics.
	WORKFLOW	Traces the flow of work requests through the server as they are placed in the queue and assigned to a thread.
	LOCK	Traces the locking and unlocking of resources.
	COMMAND	Zeke commands executed by the Zeke server or process SQR request.
	MESSAGE	Messages as received from and sent to an OpsCentral client.
	MSGCONTENT	Logs all messages received by the Zeke server and the first 10,000 bytes of each message sent to an OpsCentral client.
	MSGIN	Logs incoming Zena protocol messages.
	MSGOUT	Logs outgoing Zena protocol messages.
	PARSER	Traces XML parse callbacks as XML requests from the OpsCentral client are parsed.

	SUPPRESS	Indicates whether to suppress the specified types of trace output from the Zeke server logs.
		Note:
		Trace output always is written to the dataspace logs if the OASIS dataspace logger is active.
		Precede the value with a slash (/) to cancel suppression and write the output to the server logs.
		Examples:
		This command enables function call trace messages, and suppresses the trace output from the server logs:
		`F xxxZ620A,$SET TRACE "FUNCTION,SUPPRESS"`
		This command enables all trace messages, and writes all trace output to the server logs (i.e. disables suppression):
		`F xxxZ620A,$SET TRACE 'ALL,/SUPPRESS'`

Sample Output

```
Z6092I $SET TRACE "ALL /MEMORY"
Z6083I Trace messages issued:
Z6084I    function
Z6084I   /memory
Z6084I    thread
Z6084I    stats
Z6084I    fullstats
Z6084I    workflow
Z6084I    lock
Z6084I    suppress
Z6084I    command
Z6084I    message
Z6084I    msgcontent
Z6084I    parser
```

6 Field Descriptions

This chapter lists all Zeke data display and entry fields in alphabetical order. The information in these fields is maintained or viewed through the Zeke online facility.

These are the Zeke fields:

Field	Description	
%Actv	To display active jobs that are a certain percentage complete. The percentage is calculated by comparing the time ran with the average duration time.	
A (Assume Resource)	Indicates whether the resource can be used for a restart. Required for an event to be used for restart purposes. These are the valid values:	
	Y	Use the resource for a restart. The job will try to take the resource from an abended job, if the job is set to release the resource (RESKEEP=YES). The resource mode must be EX or ES and can be obtained by a restart/rerun.
	N	Do not use the resource for a restart.
	S	Job can assume the resource from itself only, if it abends.
Abhold	See Chapter 7, “Generation Options” on page 519.	
ABOK	ISPF only) Required. Indicates whether to display events that are in ABOK status. These are the valid values:	
	Y	Default. Display all dispatched events with an ABOK status.
	N	Do not display dispatched events with an ABOK status.
Action (Condition Code)	Code that identifies the action to take when the condition code is met. Required if the Stepname field contains a value. These are the valid values:	
	A	Flag the event as AEOJ during EOJ processing.

Field	Description	
	C	Cancel at this step and flag as AEOJ during EOJ processing.
	O	Continue processing (OK).
Action (GENOPT)	Displays the action required to reload the generation option field value.	
Active (Catalog Status)	See "Event Records" on page 443.	
Active (Resources)	Indicates whether the resource is available. These are the valid values:	
	Y	Default. Resource is available.
	N	Resource is not available.
	Required if there is an entry in Resource Name.	
Actv	ISPF only) Required. Indicates whether to display all active events currently running.	
	These are the valid values:	
	Y	Default. Display all active events.
	N	Do not display active events.
A/D	ISPF only)) Indicates how the data should be sorted in Schedule View.	
	These are the valid values:	
	A	Ascending order
	D	Descending order
Added	Displays the date the documentation or JCL was added to the Zeke database.	
Addinact	See Chapter 7, "Generation Options" on page 519.	
Aftr Time	See "Notafter" on page 472.	

Field	Description
Alert Tolerance	Valid for job events only. Specifies the value ranging from `0` through `100` to be used to calculate the acceptable range of duration times (as indicated in the Normal Range field) for the event. If duration alerts are enabled for this event, executions that run shorter or longer than the Normal Range will generate an alert. See the section on Event Activity Accounting in the *ASG-Zeke Scheduling for z/OS User's Guide* for details on how this number is used to calculate the Normal Range value. The smaller the Alert Tolerance, the more likely it is that the job will generate a duration alert. In general, this number should be set high enough to avoid frequent alerts. A low tolerance should be set for critical events; a high tolerance should be set for non-critical events whose duration is unpredictable or inconsistent. The default value is `50`. Since the Alert Tolerance, Normal Range, and Avgdur fields are interdependent, if you attempt to modify more than one field at once, the system will accept or ignore your changes based on this priority order: • Alert Tolerance • Normal Range (high) • Normal Range (low) • Avgdur For example, if you change all these fields, only the change to Alert Tolerance is accepted, and the system calculates the Normal Range values. If you change only the Normal Range low value, the system calculates the Normal Range high value and the Alert Tolerance; if you change only the Normal Range high value, the system calculates the Normal Range low value and the Alert Tolerance. **Note:** Alerts can be turned on/off for a specific event using the Enable Duration Alerts field on the EMR or for all events using the Duralert generation option.
Allowed Functions	Indicates the level of access allowed for each online function (Calendar, Documentation, Event, Options, Restart, Security, Variable, Work Center, Zeke Commands). These are the valid values: `R` Read only. **Note:** This setting is not valid for work center events.

Field	Description	
	W	Write and read.
	A	Automatically enters this function in Write mode when the operator logs on. Only one function can be defined with Auto.
Apl Id	See "Appl (Event)" on page 428.	
App	See "Appl (Event)" on page 428.	
Appl (Event)	A user-assigned code to identify the application the event is a part of (up to eight alphanumeric characters long). This field is used to sort and select events by the Report Writer, Work Center Control, Schedule View (ISPF only), and Zeke operator commands.	
Appl (Variable)	A user-assigned code to identify the application the variable is a part of (up to eight alphanumeric characters long). This field is used to sort and select variables by the Report Writer and Zeke operator commands.	
Application	See "Appl (Event)" on page 428.	
Applic Id	See "Appl (Event)" on page 428.	
Argument	(z/OS only) Argument strings to be passed to the REXX exec when it is dispatched. The string's values can be parsed into local REXX variables in the exec.	
Assume Resource	See "A (Assume Resource)" on page 425.	
Aur	See Chapter 7, "Generation Options" on page 519.	
AurIntv	See Chapter 7, "Generation Options" on page 519.	
AurMsg	See Chapter 7, "Generation Options" on page 519.	
Auto	Indicates to adds 1 to the value for number of dispatch times if the scheduled event is active. The REFRESH and ENABLE parameters are assumed. **Note:** This parameter is not valid for work center events.	
AUTO	See "Options" on page 484.	

Field	Description
Automatic Reply Element Number	Displays the Zeke-assigned number that identifies the reply to Zeke. When there are multiple replies to the same message text, Zeke issues the elements in sequence starting with the lowest number and flags the elements as used. If the message is issued more times than there are replies, the last used element is repeated. If a message is defined with only one reply, Zeke issues that reply as many times as needed.
Auto Reply	ISPF only) An asterisk (*) to indicates that auto reply segments exist for this event.
Avdur	See "Avgdur" on page 429.
Average Run Time	See "Avgdur" on page 429.
Avg	See "Avgdur" on page 429.
Avgdur	Specifies the average amount of time (in *hh*:*mm*:*ss* format) that the job takes to execute. Zeke calculates the average duration by adding the current duration to the last ten execution times, then dividing by 11. If the job has not been dispatched ten times yet, the number of times it has been dispatched is used in the calculation instead. If a job abends, the execution is not included in the average duration calculation. You can edit this field manually on the Event Master Definition screen or Event Master Record Accounting screen. If you change the Avgdur, Zeke uses the new value to recalculate the Normal Range values for the event. **Note:** If the Avgdur field is updated, the **Job ran __ Times** value is automatically reset to `1`, and the standard deviation is re-initialized to 25 percent of the average duration. Since the Alert Tolerance, Normal Range, and Avgdur fields on the Event Master Record Accounting screen are interdependent, if you attempt to modify more than one field at once, the system will accept or ignore your changes based on this priority order: • Alert Tolerance • Normal Range (high) • Normal Range (low) • Avgdur

Field	Description
	For example, if you change all these fields, only the change to Alert Tolerance is accepted, and the system calculates the Normal Range values. If you change only the Normal Range low value, the system calculates the Normal Range high value and the Alert Tolerance; if you change only the Normal Range high value, the system calculates the Normal Range low value and the Alert Tolerance. **Note:** If you change the Clear Duration Stats field to `Y`, Avgdur value resets to `00:00:00`.
Avrge Dur	See "Avgdur" on page 429.
Batsec	See Chapter 7, "Generation Options" on page 519.
Bimappl	See Chapter 7, "Generation Options" on page 519.
Bimpasw	See Chapter 7, "Generation Options" on page 519.
Bimuid	See Chapter 7, "Generation Options" on page 519.
Blocks Free	Displays the number of free blocks in the database. This number is obtained by subtracting the number of used blocks from the total number of blocks available.
Blocks in Use	Displays the number of blocks that are being used by Zeke records.
% Blocks Used	Displays the percentage obtained when you divide the number of used blocks by the total number of database blocks available.
by {user ID}	Displays the user ID of the last operator who updated the record.
Bypjob	See Chapter 7, "Generation Options" on page 519.
Ca	See "Allowed Functions" on page 427.
Cal	See "Allowed Functions" on page 427. See "Calendar Id" on page 431.
Calcmem	See Chapter 7, "Generation Options" on page 519.
Calctap	See Chapter 7, "Generation Options" on page 519.
Calendar (Audit)	Required. Indicates whether you want to track and log changes to calendar records. These are the valid values:

Field	Description	
	N	Default. Do not track and log calendar changes.
	Y	Track and log calendar changes.
Calendar (Security)	See "Allowed Functions" on page 427.	
Calendar End Date	Displays the last date of the calendar. For user accounting calendars, Zeke calculates this date based on the start date, the number of days in each period, and the number of slack days. This date must be contiguous with the start date of the calendar for the next year.	
Calendar Expire Date	Optional. Specifies the date (in *mm/dd/yyyy* format) that the calendar expires. On the date of expiration, the calendar is deleted when you run the schedule load for that day. You should set the Calendar Expire Date for six days beyond the actual desired expiration date to ensure proper calculation of OCCURS hits.	
Calendar Id	Specifies the user-assigned calendar ID (up to eight alphanumeric characters long). Do not use the same name for different calendars, even if the calendar type and year are different. On the EMR screen, this is the calendar ID to use when scheduling the event, and is required. The calendar must exist before the SCHEDULE function runs. The default calendar ID is A. Special calendars are not valid in this field.	
Calendar Start Date	The first date of the calendar according to the fiscal start month, in *mm/dd/yyyy* format.	
Calendar Type	Indicates the type of calendar: These are the valid values:	
	STD	Standard
	SPC	Special (ISPF only)
	SPEC	Special (native only)
	USER	User Accounting (native only)
	USR	User Accounting (ISPF only)
Calendar Year	See "Year (Calendar)" on page 516.	
Calid	See "Calendar Id" on page 431.	
Capacity	See "Event Records" on page 443.	

Field	Description
Caps	If your WHEN condition contains lower case characters and you want the system to convert them to upper case, you can set the Caps option to `ON`. If you want your clause to remain in lower case or mixed case, set Caps to `OFF`.
Catalog Blocks	Displays the total number of database blocks. The allocated database size.
Catalog Created	Displays the date the database was created.
Cause to Rerun	ISPF only) Optional. Specifies the reason to run the job again (up to 60 alphanumeric characters long).
Chgval	See Chapter 7, "Generation Options" on page 519.
Class *(Job EMR)*	Optional. Valid for job events only. Specifies up to six classes for the event. The valid values range from `A` through `Z`, and `0` through `9`. See the section, "Initiator Processing" in Chapter 6, "Resources" in the *ASG-Zeke Scheduling for z/OS User's Guide* for more information. **Note:** If DispSel is set to `Y`, you also can define up to 36 unique, single-character job classes to be treated as exceptions (as if DispSel is set to `N`). See "Class (Job Class Capacity)" on page 432 for details. See the *ASG-Zeke Scheduling for z/OS User's Guide* for more information on initiator processing and selection.
Class *(Job Class Capacity)*	Optional. If DispSel is set to `Y`, these job classes are treated as exceptions (as if DispSel is set to `N`). Defining job class exceptions enables Zeke-controlled jobs to be submitted to JES without having to wait for an available initiator (for example, even if initiators are managed by Workload Manager). You can specify up to 36 unique, single-character job classes. The valid values range from `A` through `Z` and `0` through `9`. **Note:** You must issue the ZRELOAD INIT command or cycle Zeke using ZKILL COLD or TRACK for the changes to become effective. You also can set limits on the number of jobs that Zeke can submit for each of these classes. This controls the number of Zeke-submitted jobs in the JES queue at the same time, which increases the potential for the jobs to be execute in closer accordance with your Zeke-defined dispatching priorities. See "Max (Job Class Capacity)" on page 466 for details.

Field	Description
Class *(Partition)*	Required. Single-character partition class.
Class *(REXX)*	(z/OS only) OASIS ECF exec class with which the REXX exec is associated. The valid classes are `A` through `Z` and `0` through `9`.
Class ID (Security)	Specifies the user-assigned class ID for this operator ID. The class defines the online screens and Zeke commands the operator can have access to. The valid values are `A` through `Z`. The default class is `A` when the database is created. On the Directory of Command Classes screen, a list of defined class IDs is also listed.
Clause (partial)	Displays the WHEN condition (or SET clause for WORK events) for this version of the event. If the clause is too long to fit on the screen, enter the Browse line command to view the entire clause on the Event Master Record Function screen. See the WHEN field description in this chapter for more information on WHEN conditions, or to the SET field description for more information on SET clauses.
Clear Duration Stats	Valid for job events only. Set this field to `Y` to reset the job duration statistics for this event. Specifically, this resets these values: • Avgdur to `00:00:00` • Job ran __ Times to `1` • Normal Range to `00:00:00` - `00:00:00` If duration alerts and/or duration failures are enabled for this event, these will not occur until the job runs at least the number of times indicated by the Durcount generation option.
Cls	See "Class ID (Security)" on page 433.
Cmdcons	See Chapter 7, "Generation Options" on page 519.
Cmsftype	See Chapter 7, "Generation Options" on page 519.
Cnt (Resource)	Specifies the number (up to three digits long) of specified resources required by the event. The default value is `001`. This parameter is required if Resource Name field contains a value.
CNT (Schedule Times)	See "Times" on page 504.

Field	Description
Code (SCOM Event)	Required. Enter the code that indicates the command type in the space to the right of the Code field. Then, enter up to 60 alphanumeric characters of system commands and/or responses per line (unlimited lines). Only the first line is required. Command lines 001 through 007 display at the bottom of the screen. To display additional command lines, place the cursor on any Code field (except the first) and press Enter. Return to line 001 by positioning the cursor on the title line and press Enter. These are the valid values: C — System command R — System response (VSE only) Z — Zeke command V — VM command P — VSE/POWER command (VSE only) *Batch Parameters:* All under EVENT function—SCOM1, SCOM2, SCOMx, etc. On the ESI Customization - X1FOFSEL screen, this field is optional. Enter $ next to the appropriate Command Code field to add the code to the resource name format.
Color	Indicates the color desired for the listed item. The colors are displayed near the bottom of the screen. You only have to enter the first letter of the color. These are the valid values: R — Red B — Blue G — Green Y — Yellow W — White P — Pink T — Turquoise
Command Code	ISPF only)) See "Code (SCOM Event)" on page 434.
Commctl	See Chapter 7, "Generation Options" on page 519.

Field	Description
Comment Line 1 though Comment Line 6	Specifies the description of the work center event activity (up to 60 alphanumeric characters long). *Batch Parameters:* All under the EVENT function—LINE1, LINE2, LINE3, LINE4, LINE5, LINE6
Comp.code	Any completion codes that were returned.
Condcode	Displays the condition codes that were returned.
Cond Codes	ISPF only)) An asterisk (*) indicates that condition codes exist for this event.
Condrdv	See Chapter 7, "Generation Options" on page 519.
Condrlb	See Chapter 7, "Generation Options" on page 519.
Condrver	See Chapter 7, "Generation Options" on page 519.
Control	Required. Indicates whether this event is tracked as a Zeke-controlled job. Zeke-controlled jobs are tracked throughout the entire execution. These are the valid values: `Y` Default. Zeke recognizes this event as a Zeke-controlled job. `N` Zeke does not recognize this event as Zeke-controlled and marks the job as DONE upon dispatch. `NX` Zeke recognizes the event as a non-executable Zeke-controlled event. Non-executable events are scheduled like any other event, and are useful as predecessors to other events. A non-executable event is never submitted to JES for JCL execution. After dispatch, the event status automatically changes to indicate success and any dependent events are triggered. **Note:** An event you want to remove from an intricate event flow can simply be marked as non-executable as an alternative to changing the event flow, which typically means having to delete the event and then modify the WHEN clauses for all of the deleted event's successors.
Copy to System	Optional. Specifies the name of the system to copy the ESI class definition record to for external security use (up to eight alphanumeric characters long). This field is used if two or more systems share the same database and have different requirements for process options, external class names, and/or resource name formats.

Field	Description
Cputime	Displays the amount of CPU time (in *hh*:*mm*:*ss* format) the job took to run.
Count (Partition)	VSE only. For dynamic partitions only; indicates the number of dynamic partitions you want Zeke to control.
Count (Resource)	See "Cnt (Resource)" on page 433.
Curr Value	The current variable value. A character value can be up to 64 bytes long; a numeric value can be from -2,147,483,647 to +2,147,483,647.
Current Format	ISPF only)) An asterisk (*) indicates this is the currently used format for displaying dates in Schedule View.
Customer ID	Displays the unique number that identifies your company. (This ID is supplied by ASG when required.)
D	See "Allowed Functions" on page 427.
Data-Name Records	See "Event Records" on page 443.
Data Name	See "Variable" on page 508.
Dataset Name	Specifies the name of the dataset (up to 44 alphanumeric characters long).
Datasub	See Chapter 7, "Generation Options" on page 519.
Date (Variable)	Displays the date the variable was set.
Date Added	Displays the date the operator ID was added to the Zeke database.
Date Last Accessed/ Used	Displays the date the calendar was last used by the SCHEDULE function.
Date Last Updated	Displays the date the operator ID was last updated.
Date Range	Optional. Specifies a date range in *mm/dd/yyyy* format. In the first field, you specify the starting date of the range you want to match. In the second field, you specify the ending date of the range you want to match.
Date/Time	Displays the date and time the variable value was set.
Days and Months	Optional. Specifies the days and months are listed across the top and down the side of the screen creating a table. Enter an asterisk (*) for each of the days that you want the job to run.

Field	Description
Deac	See "Event Records" on page 443.
DefDelOJ	See Chapter 7, "Generation Options" on page 519.
Defdprty	See Chapter 7, "Generation Options" on page 519.
Defdspcl	See Chapter 7, "Generation Options" on page 519.
Defjcl	See Chapter 7, "Generation Options" on page 519.
Defjprty	See Chapter 7, "Generation Options" on page 519.
Defopid	See Chapter 7, "Generation Options" on page 519.
Defpltfm	See Chapter 7, "Generation Options" on page 519.
DefSysId	See Chapter 7, "Generation Options" on page 519.
Desc (Event)	The user-assigned description of the event (up to 44 alphanumeric characters long).
Desc (Variable)	Optional. Specifies the user-assigned description of the variable (up to 44 alphanumeric characters long).
Desc2 (Event)	Specifies the second line of the user-assigned event description (up to 44 alphanumeric characters long).
Desc2 (Variable)	Specifies the second line of the user-assigned variable description (up to 44 alphanumeric characters long).
Description (Date Selection)	ISPF only)) Displays the different date formats you can select for displaying dates on the Schedule View screen.
Description (GENOPT)	Specifies the user-defined description of the local or `*GLOBAL` GENOPT or the static description of the `*ACTIVE` GENOPT.
Description (System)	Optional. Specifies the description of the system or system pool (up to 20 alphanumeric characters long).
Description (User Color)	ISPF only)) Shows the types of data displayed on the User Color select screen. Screen Title — First line of the screen. Field and Column — Labels that identify the fields. Field names and Heading column headings are treated the same, whether the fields are arranged in a column or placed next to the field name that applies to them.

Field	Description	
	Normal Text	Descriptive or narrative text that usually explains or introduces other information displayed on the screen. For example, in the phrase 'D - Delete line,' 'Delete line,' which is the explanation, is normal text, while 'D,' which is the entry, is accented.
	Accented Text	Descriptive or narrative text that is highlighted for emphasis, such as the line command letter to enter in the Select field. For example, in the phrase 'D - Delete line,' 'D,' which is the entry, is accented.
	Normal Output Data	Displayed information that has previously been entered or has been calculated by the system.
	Accented Output Text	Displayed information that has previously been entered or has been calculated by the system, and has been highlighted for emphasis.
	Normal Input Data	Fields available for entry.
	Accented Input Data	Fields available for entry; highlighted for emphasis.
	Input Field in Error	(Not used)
Detail	Displays an asterisk (*) in column to indicate that time range information exists for this /initiator. Place the cursor on the initiator ID and press F10 to display the Time Range screen. If no detail exists, the initiator is available 24 hours a day, 7 days a week.	
Dispatch Date *mm/dd/yyyy* at *hh:mm:ss*	Displays the last three dates and times the event was dispatched.	
Dispatched *x* Times	Displays the number of times the job has been dispatched.	
Disp Class	Optional. Specifies the dispatch class (up to two digits long) to match.	
Dispatching Class	See "Disp Class" on page 438.	
Dispdly	See Chapter 7, "Generation Options" on page 519.	
DispSel	See Chapter 7, "Generation Options" on page 519.	
Doc	An asterisk (*) indicates that documentation segments exist for this event.	

Field	Description
Doc (Security Access)	See "Allowed Functions" on page 427.
Document	See "Allowed Functions" on page 427.
Documentation	See "Allowed Functions" on page 427.
Documentation Note Information	Optional. Enter any information related to this event, such as instructions to operations personnel. You can add up to ten lines of information (up to 60 alphanumeric characters per line).
Documentation Scratch Pad Info	Optional. Enter any information related to the event. You can add up to ten lines of information (up to 60 alphanumeric characters per line).
Documentation Text Information	Optional. Specifies the text information as entered in the /EVENT documentation function (up to 60 alphanumeric characters long).
DONE	ISPF only)) Required. Indicates whether to display events that have already been dispatched. These are the valid values: `Y` Default. Display all dispatched events with a DONE status. `N` Do not display dispatched events. **Note:** If you set DONE to `Y`, Zeke displays all dispatched events (including those with an ABOK, Fail, FBOK, or FSucc status, regardless of how those fields are set). If you set DONE to `N`, as well as set one of the specific status fields (i.e., ABOK, Fail, FBOK, or FSucc) to `Y`, then Zeke displays only the events with that particular status.
Download Status	Download status of the job event. These are the valid values: `NOTSNT` Event has not yet been downloaded to Zeke Agent. `NOSUPP` The event was not downloaded because it has an attribute not supported for downloaded schedules, such as logical resources or a condition code record. `ERROR` An internal error occurred during schedule download to target.

Field	Description	
	SENT	Job event has been successfully downloaded to Zeke Agent.
	PENDING	An update to the downloaded schedule record is pending.
	STALEZ	Event status is mismatched between Zeke and Zeke Agent. The Zeke status is out-of-date.
	STALEA	Event status is mismatched between Zeke and Zeke Agent. The Zeke Agent status is out-of-date.
	??xx??	Unknown status. Contact ASG Customer Support for assistance.
DP	See "Dprty" on page 440.	
Dprty	Required. Specifies the dispatch priority number ranging from 0 (highest) through 99 (lowest). The default value is 50.	
DRL	Optional. Specifies the appropriate disaster recovery level for the event. The valid values range from 1 through 99. This field is used by the Report Writer, work center control, and the SCHEDULE function to sort and select events similar to the application and group IDs.	
Dscltrig	See Chapter 7, "Generation Options" on page 519.	
Dsntrig	See Chapter 7, "Generation Options" on page 519.	
DSPBatch	See Chapter 7, "Generation Options" on page 519.	
DSPIndex	See Chapter 7, "Generation Options" on page 519.	
DSPSched	See Chapter 7, "Generation Options" on page 519.	
Dsptch Class	Optional. Specifies the class or class list for the event (up to six alphanumeric characters long).	
Dur	(Display) Format *hh*:*mm*:*ss*. The number of hours, minutes, and/or seconds it took the job to run.	
Duralert	See Chapter 7, "Generation Options" on page 519.	
Durcount	See Chapter 7, "Generation Options" on page 519.	
Durfail	See Chapter 7, "Generation Options" on page 519.	
Dynsmf	See Chapter 7, "Generation Options" on page 519.	

Field	Description
E	See "Allowed Functions" on page 427.
E (Resource Fail)	Indicates whether to release the resource at job failure (AEOJ). Required if there is an entry in the Resource Name field. These are the valid values: `R` Default. Release the resource. `K` Keep the resource if the event abends. This resource can be obtained by a restart/rerun event.
Early	See "Early Time" on page 441.
Early Time	Specifies the earliest time (in *`hh:mm`* format) that the event can be dispatched. The valid values range from `00:00` through `47:59`. If the time is greater than `24:00`, Zeke considers to event to require dispatching on the next day. **Note:** It is possible for Zeke to dispatch an event at its early time; however, Zeke still dispatches the event in sequence according to its schedule time.
Edate	See "Calendar End Date" on page 431.
EDBindex	See Chapter 7, "Generation Options" on page 519.
Effective as of	Optional. Specifies the date the reply becomes effective, in *`mm/dd/yyyy`* format.
Effective until	Optional. Specifies the date the reply is no longer effective, in *`mm/dd/yyyy`* format.
ENABLE	See "Options" on page 484.
ENABLE Duration Alerts	Valid for job events only. This field can be used to override the Duralert generation option for a particular event. (Duralert indicates whether a console message and OpsCentral alerts should be issued if a job runs longer or shorter than the acceptable range of duration times.) These are the valid values: `Y` Enable duration alerts for this event. Zeke issues message Z0319W. `N` Disable duration alerts for this event. (blank) Default. Use the Duralert generation option value.
Ename	The user-assigned event name (up to 12 alphanumeric characters long).

Field	Description
End (EMR Accounting)	Displays the end date, in *mm/dd/yyyy* format, and end time, in *hh:mm:ss* format, of the last execution of the job during which the job ran and completed (SUCCESS, FAIL, F/SUCC, F/FAIL, or FAIL LONG). If the job has never dispatched, this field does not appear on the screen.
End Date	See "Calendar End Date" on page 431.
Ending Procname	Optional. Specifies the name of the last procedure to execute on a restart/rerun (up to eight alphanumeric characters long).
Ending Stepname	Optional. Specifies name of the last step to execute on a restart/rerun (up to eight alphanumeric characters long).
English When Conditions	Optional. Specifies the WHEN condition for the event (up to 1,360 alphanumeric characters long). To add an AND/OR condition to the end of the existing condition, tab to the end of the line and type the new information. See the *ASG-Zeke Scheduling for z/OS User's Guide* for more information on viewing and maintaining WHEN conditions.
EOJ CC	Specifies the maximum condition code (*nn*) allowed for any step in the job.
Eojwake	See Chapter 7, "Generation Options" on page 519.
ESI Actv	See Chapter 7, "Generation Options" on page 519.
Ev	See "Allowed Functions" on page 427.
Event	Specifies the Zeke-assigned event number (*nnnnnn*). See also "Allowed Functions" on page 427.
Event Master Record (Audit)	Required. Indicates whether you want to track and log all changes to EMRs. These are the valid values: `N` Default. Do not track and log EMR changes. `Y` Track and log all EMR changes.
Event Name	See "Ename" on page 441.
Event No.	See "Event" on page 442.
Event Number	See "Event" on page 442.

Field	Description
Event Records	(Display) Displays the number of Zeke records that are in each of these states: • Active • Inactive • Used • Unused Also shows the total potential number of Zeke records (capacity) based on the database size, and the percentage of records that are in use.
Event Status/Reason Code	(Display) Displays the processing status of the event, whether it is TIMEOK and/or WHENOK, and whether it has been dispatched. The reason code is displayed to the right of the status, and indicates why the event is waiting to execute (if applicable). **Note:** Status codes are displayed in Schedule View, in ZDISPLAY output, on the Event Master Record Accounting screen, and on the LIST PLAN reports. Some codes are displayed differently in different parts of the system (e.g., F/FAIL versus F/FL), but have the same meaning. For a list of all possible statuses and reason codes, see the Scheduling chapter in the *ASG-Zeke Scheduling for z/OS User's Guide*.
Event Totals by Type	(Display) Displays the total number of events (grouped by event type) owned by this system.
Event Type	Displays the specific type of event assigned to this event when it was defined. See "Event Types" on page 443 for a list of all event types.
Event Types	Optional. Enter any character, except a space, next to the code for the event type(s) you want to select. These are the valid event types: Job — Job event Msg — Message to system operator Pcom — POWER command (VSE only) Work — Work center event Vcom — VM command Scom — System command Zcom — Zeke command

Field	Description
	REXX (z/OS only) REXX exec
Evt	See "Event" on page 442.
Evt Name	See "Ename" on page 441.
Evt Type	See "Event Type" on page 443.
Exp	See "Expire (EMR)" on page 444.
Expire (EMR)	The date the event expires, in MMDDYYYY or DDMMYYYY format. When the SCHEDULE function runs on or after this date with the ACTIVATE parameter, the event is automatically deleted. If a schedule forecast is run for a date that is after the expiration date, a message informing you of the expired event is displayed. The event is not deleted.
External Class	Optional. Specifies the class name that is used for SAF calls (up to eight alphanumeric characters long). The class specified must be defined to the external security product.
External- Security Class Def (Audit)	Required. Indicates whether you want to track and log all changes to external security class definition records. These are the valid values: `N` Default. Do not track and log changes to external security class definition records. `Y` Track and log all changes to external security class definition records.
Fail	ISPF only) Required. Indicates whether to display events that in Fail (AEOJ) status. See also "E (Resource Fail)" on page 441. These are the valid values: `Y` Default. Display all dispatched events with a Fail status. `N` Do not display dispatched events with a Fail status.
Fail if short or long	Valid for job events only. This field can be used to override the Durfail generation option for an event. (Durfail indicates whether Zeke should fail jobs that run longer or shorter than the acceptable range of duration times. Marking these jobs as failed prevents them from triggering successor jobs.)

Field	Description
	Note: This option does not cause a job to be cancelled. When a job ends, if its duration fell outside the Normal Range, Zeke marks the job as failed and issues message Z8T02I. If a job abends or fails due to a condition code record, Zeke does not fail the job because of its duration. Zeke fails the job due to its duration only if it otherwise would have been marked as successful. These are the valid values: `Y` Fail this job if it runs longer or shorter than the acceptable range of duration times. `N` Do not fail this job if it runs longer or shorter than the acceptable range of duration times. (blank) Default. Use the Durfail generation option value.
Fairmod	See Chapter 7, “Generation Options” on page 519.
Fairopn	See Chapter 7, “Generation Options” on page 519.
Fairrec	See Chapter 7, “Generation Options” on page 519.
FBOK	ISPF only) Required. Indicates whether to display events that have been dispatched with a FBOK status. These are the valid values: `Y` Default. Display all dispatched events with an FBOK status. `N` Do not display dispatched events with an FBOK status.
Field Description (ESI Security)	A brief representation of a particular element that can be included in the resource name format (up to 60 alphanumeric characters long). An element can be any of these: • The value of a data field. • Selected positions from the value of an eligible data field. For example, positions 3 and 4 of the Jobname field from the event master record. • Any arbitrary string of one to eight characters (a literal value). • A delimiter character. Most elements are fields in the record being secured. However, they can be structures associated with the Zeke database, or constants applicable to the element. The most common elements are described below:

Field	Description	
	Catalog ID	The 8-byte unique catalog identifier, determined at the time the catalog is created. The catalog ID appears after the word CATID in the last line of output from a ZID command. Once created, the catalog ID cannot be changed, even if the catalog is restored from a backup dataset.

	Command verb	The command verb issued. For example:
		`Z$CMD Class–ZID`
		`ZDISPLAY`
		`ZALTER`
		`Z$SET Class–SET`
		`CDATE`

	Command Code	Code specified in an SCOM. See "Code (SCOM Event)" on page 434 for more information.
	Command Text	The full text of an entered command.

Field	Description	
	Function	The name of the online function to which access is being requested. Potential values: • LOGON—Logon to the Zeke online system • LOGOFF—Exit the Zeke online system • EVENT—EVENT Primary Menu option • ZCOM—ZCOM/SCHEDULE VIEW Primary Menu option • CALENDAR—CALENDAR Primary Menu option • OPTIONS—OPTIONS Primary Menu option • WORK—WORK Primary Menu option • SECURITY—SECURITY Primary Menu option • DOCUMENT—DOCUMENT Primary Menu option • VARIABLE—VARIABLE Primary Menu option • RESTART—RESTART Primary Menu option • GWSSEC—Zeke/GWS access command • BLOCK—Special function to view Zeke catalog blocks • MEMORY—Special function to view memory • EVENTAD*x*—Special function to control access to the Event ADD screen by event type. Actual values: • EVENTADJ—Job event ADD screen • EVENTADM—Message event ADD screen • EVENTADP—Pcom event ADD screen • EVENTADC—Work center event ADD screen • EVENTADV—Vcom event ADD screen • EVENTADZ—Zcom event ADD screen • EVENTADS—Scom event ADD screen • EVENTADR - REXX event ADD screen • EVENTADT - Task event ADD screen
	Record Type	Sub-record type to which access is requested. For example, an EMR has several types of subordinate records (e.g., JCL, DOC, COND CODEs, etc.)

Field	Description
	Potential values are: EMR—Main Event Master Record • OCCURS—Occurs record • WHEN—When record • JCL—Zeke JCL • AUTOREPL—Zeke auto replies • CONDCODE—Zeke condition code checking • RESOURCE—Zeke event resource definitions • DOC NOTE—Documentation NOTE • DOC SPAD—Documentation SCRATCH PAD • DOC TEXT—Documentation TEXT • DOC DSET—Documentation DATASET SQR—Main Schedule Queue Record • JCL—Zeke JCL • RESOURCE—Zeke event resource definitions • WHEN—When record CALENDAR—Main Calendar record • DOC NOTE—Documentation NOTE • DOC SPAD—Documentation SCRATCH PAD • DOC TEXT—Documentation TEXT VAR—Main Variable record • DOC NOTE—Documentation NOTE • DOC SPAD—Documentation SCRATCH PAD • DOC TEXT—Documentation TEXT
Source	This is the source of the request that caused the security call to be made. For example, if you issue a command from the console, the value of the source field is CONSOLE.

Field	Description
	These are the valid values: • CONSOLE—Request from the operator's console. • ONLINE—Request from an online system (CICS, TSO, ISPF, etc.). • BATCH—Request from a batch program or job (e.g., ZEKE batch utility, ZEKESET program, etc.) • ZEKECMD—Request from the ZEKECMD subroutine. • ZEKEVAR—Request from the ZEKEVAR subroutine.
Subsystem Name	This is the OASIS subsystem that this particular Zeke is running under. Multiple subsystems allow the user to run more than one Zeke on the same operating system. Its value is shown on the third line of the response to the ZID command.

This table is a complete list and description of all eligible elements for every class.

Note:

See the Security chapter in the *ASG-Zeke Scheduling for z/OSUser's Guide* for more details.

—Internal Classes—

Z$CATAL — Secure access to database when accessed by a global function.

```
Record Length Elements
     001      Delimiter character
     1-8      Literal value ===> _________ (quoted)
     001      Security class
     008      Zeke internal catalog ID*
     004      Subsystem name*
     007      Source*
```

Note:

The Z$CATAL format is fixed and cannot be changed.

Z$CLASS — Secure access to CLASS records which are used in internal security.

Field	Description
	Record Length Elements 001 Delimiter character 1-8 Literal value ===> _____ (quoted) 001 Security class 008 Zeke internal catalog ID* 004 Subsystem name* 007 Source*
Z$CMD	Secure access to operator commands.
	Record Length Elements 001 Delimiter character 1-8 Literal value ===> _____ (quoted) 016 Command Verb* 008 Zeke internal catalog ID* 004 Subsystem name* 007 Source*
Z$CND	Secure access to Calendar Records.
	Record Length Elements 001 Delimiter character 1-8 Literal value ===> _____ (quoted) 008 Calendar name 004 Year 003 Calendar type 008 Record type* 008 Zeke internal catalog ID* 004 Subsystem name* 007 Source*
Z$ECDR	Secure access to ESI Class Definition Records.
	Record Length Elements 001 Delimiter character 1-8 Literal value ===> _____ (quoted) 008 System name 008 Internal class name 008 External class name 008 Zeke internal catalog ID* 004 Subsystem name* 007 Source*
Z$EMR	Secure access to Event Master Records & subordinate records.

Field	Description
	Record Length Elements 001 Delimiter character 1-8 Literal value ===> _____ (quoted) 008 User ID 008 Application ID 003 Group ID 012 Event name 008 Job name (JOB EVENTS ONLY) 008 System name 004 Event type 008 Record type* 008 Target 008 Platform 007 Source* 002 Disaster recovery level 008 Zeke internal catalog ID* 004 Subsystem name* 048 JCL member info 004 JCL source 008 Calendar ID 002 Dispatch priority 002 Job card priority 006 Init/Part class 001 Template 060 Description 060 Description2 016 Command verb* 005 Event number (NOT RECOMMENDED)
Z$GOPT	Secure access to Generation Option records.
	Record Length Elements 001 Delimiter character 1-8 Literal value ===> _____ (quoted) 008 System name 008 Zeke internal catalog ID* 004 Subsystem name* 007 Source*
Z$NAME	Secure access to Name & Address records.
	Record Length Elements 001 Delimiter character 1-8 Literal value ===> _____ (quoted) 008 Zeke internal catalog ID* 004 Subsystem name* 007 Source*
Z$ONLINE	Secure access to ONLINE functions.

Field	Description
	Record Length Elements 001 Delimiter character 1-8 Literal value ===> _____ (quoted) 008 Function* 008 Zeke internal catalog ID* 004 Subsystem name* 007 Source*
	Z$OPER Secure access to OPERATOR records.
	Record Length Elements 001 Delimiter character 1-8 Literal value ===> _____ (quoted) 008 Zeke operator ID 008 Zeke internal catalog ID* 004 Subsystem name* 007 Source*
	Z$PASS Secure access to operating passwords.
	Record Length Elements 001 Delimiter character 1-8 Literal value ===> _____ (quoted) 008 Zeke internal catalog ID* 004 Subsystem name* 007 Source*
	Z$PINT Secure access to partition definitions.
	Record Length Elements 001 Delimiter character 1-8 Literal value ===> _____ (quoted) 008 System name 008 Zeke internal catalog ID* 004 Subsystem name* 007 Source*
	Z$POOL Secure access to POOL definition records.
	Record Length Elements 001 Delimiter character 1-8 Literal value ===> _____ (quoted) 008 Pool ID 008 Zeke internal catalog ID* 004 Subsystem name* 007 Source*
	Z$RESRC Secure access to RESOURCE definition records.
	Record Length Elements 001 Delimiter character 1-8 Literal value ===> _____ (quoted) 008 System name 044 Resource name 008 Zeke internal catalog ID* 004 Subsystem name* 007 Source*

Field	Description
Z$SCHED	Secure access to SCHEDULE function.

```
Record Length Elements
     001      Delimiter character
     1-8      Literal value ===> _________ (quoted)
     016      Command verb*
     008      System name
     008      Application ID
     003      Group ID
     008      User ID
     002      Disaster recovery level
     008      Zeke internal catalog ID*
     004      Subsystem name*
```

Z$SET	Secure access to ZEKESET commands.

```
Record Length Elements
     001      Delimiter character
     1-8      Literal value ===> _________ (quoted)
     016      Command verb*
     008      Object keyword
     008      Zeke internal catalog ID*
     004      Subsystem name*
```

Z$SIM	Secure access to SIMULATION function.

```
Record Length Elements
     001      Delimiter character
     1-8      Literal value ===> _________ (quoted)
     016      Command verb*
     008      User ID
     008      Application ID
     003      Group ID
     008      System name
     008      Zeke internal catalog ID*
     004      Subsystem name*
```

Z$SQR	Secure access to Schedule Queue records.

Field	Description

```
Record Length Elements
     001      Delimiter character
     1-8      Literal value ===> _________ (quoted)
     008      User ID
     008      Application ID
     003      Group ID
     012      Event name
     008      Job name
     004      Event type
     008      Record type*
     007      Source*
     008      System name
     008      Target
     008      Platform
     008      Zeke internal catalog ID*
     048      JCL member info
     004      JCL source
     002      Dispatch priority
     002      Job card priority
     004      Subsystem name*
     016      Command verb*
     006      Init/Part class
     005      Event number (NOT RECOMMENDED)
```

Z$VAR Secure access to variables.

```
Record Length Elements
     001      Delimiter character
     1-8      Literal value ===> _________ (quoted)
     008      User ID
     008      Application ID
     003      Group ID
     016      Variable name
     008      Record type*
     007      Source*
     004      Subsystem name*
     016      Command verb*
     060      Description
     060      Description2
     008      Zeke internal catalog ID*
```

Z$XCOM Secure access to individual SCOM lines.

```
Record Length Elements
     001      Delimiter character
     1-8      Literal value ===> _________ (quoted)
     001      Command code*
     060      Command text*
     008      Zeke internal catalog ID*
     004      Subsystem name*
     007      Source*
```

Z$DOWNLD Secures access to the schedule download agents list.

```
Record Length Elements
     001      Delimiter character
     1-8      Literal value ===> _________ (quoted)
     008      Zeke internal catalog ID*
     004      Subsystem name*
     007      Source*
```

Field	Description
	Z$ACCESS — Secures use of generic selection criteria (for application ID, group ID, system ID, or user ID), use of the CLEAR keyword with the SCHEDULE function, and whether selected EMR fields are locked (SECULOCK=E).

	`Record Length Elements`
	`001 Delimiter character`
	`1-8 Literal value ===> _____ (quoted)`
	`064 Item to be accessed`
	`008 Zeke internal catalog ID*`
	`004 Subsystem name*`
	`007 Source*`

	* Refer to page 446 to page 449 for detailed descriptions of these elements.
Field Description (SView Display)	ISPF only) Description of the data to use as the sort on the Schedule View screen.
Fiscal Start Month	Optional. Specifies the first month of your fiscal year (if you want your calendar to follow the fiscal year). The valid values range from `01` (January) through `12` (December).
Force Job Done	ISPF only) Optional. Enter any character beside this field to force the job to SUCCESS status regardless of the current job status. All applicable WHEN conditions are updated when the event goes to Success status. Jobs forced to Success status are flagged FSucc (Forced Success).
Force Upper	Indicates whether to force the variable value string entered in the Current Value field to upper case (if the string contains alpha characters). These are the valid values: `Y` — Forces the Current Value string to uppercase. `N` — Keeps the case of the Current Value exactly as entered. You specify this code if you want to allow mixed-case values for variable substitution.
Format	ISPF only) A list of format choices for displaying dates in Schedule View.
Freq	Valid for recurring events only. Specifies the amount of time (in *`hh:mm`* format) that the event must wait before running again, The valid values range from `00:00` through `47:59`. To determine the next schedule time, Zeke adds this value to the current system time or the current schedule time, depending on the Freqcalc value. **Note:** ASG recommends that you specify a Freq time and/or a WHEN condition for permanent and recurring events.

Field	Description
	If Freq is set to `00:00`, then the recurring or permanent event is marked as time-satisfied immediately after each completion. As soon as the WHEN condition (if any) also is satisfied, Zeke dispatches the event. **Note:** For a permanent event, when the next schedule time passes 24:00, Zeke sets the run date to the current date.
Freqcalc	Optional. For recurring events only. Indicates how to calculate the next dispatch time. These are the valid values: `S` Default. Reschedule the event by the actual schedule time, no matter when the event was dispatched and completed. For example, an event is scheduled to run at 10:00 and then every two hours after that, two more times. The second and third events are still scheduled for 12:00 and 2:00 regardless of the completion time of the first. `C` Reschedule the events based on the completion time of the first one. For example, an event is scheduled to run at 10:00 and then every two hours after that, two more times. If it is completed at 10:15, the second event is scheduled for 12:15. Then, when it is completed, the third event is scheduled for two hours later.
Frequency Calc	See "Freqcalc" on page 456.
FSucc	ISPF only) Required. Indicates whether to display events that have been dispatched with a Forced Success status. These are the valid values: `Y` Default. Display all dispatched events with an FSucc status. `N` Do not display dispatched events with an FSucc status.
Full Job Name	ISPF only) Use to display the entire 30-character jobname on the Schedule View screen. On z/OS, the first eight characters are from the name on the JOB card and the rest of the characters are blanks.
Function	Displays the status of the event (Success or NotDone).
Functions	See "Allowed Functions" on page 427.

Field	Description
Genopt (Audit)	Required. Indicates whether you want to track and log all changes to the Zeke generation options. These are the valid values: `N` Default. Do not track and log generation option changes. `Y` Track and log all generation option changes.
Group (Event)	Optional. A user-assigned code to identify the group the event is a part of (up to three alphanumeric characters long). This field can be used as a subset of the application ID and to sort and select events by the Report Writer, Work Center Control, Schedule View (ISPF only), and Zeke operator commands.
Group (Variable)	A user-assigned code to identify the group the variable is a part of (up to three alphanumeric characters long). This field can be used as a subset of the application ID and to sort and select variables by the Report Writer and Zeke operator commands.
Group Id	See "Group (Event)" on page 457. See "Group (Variable)" on page 457.
Grp	See "Group (Event)" on page 457.
Grp Id	See "Group (Event)" on page 457.
GUI Serv	See Chapter 7, "Generation Options" on page 519.
H (Resource Hold)	Indicates whether to hold an available resource. Required if there is an entry in the Resource Name field. These are the valid values: `N` Default. Do not hold the resource. `Y` Hold the resource if it is available and is in the correct mode. However, the resource can be "stolen" by another event with a higher dispatch priority.
High	The highest condition code (up to four digits long) to be included in the range. This field is required if the condition code *range* (RA) operator is used; otherwise, leave this field blank.
Hilite	Indicates the attribute desired for the listed item. The attributes are displayed at the bottom of the screen. You only have to enter the first letter of the attribute. `Reverse` Displays the text on a solid background of the color specified in the Color column

Field	Description	
	`Uscore`	Underlines the text.
	`Blink`	Displays blinking text.
Hit	Displays the date the event is scheduled to occur, in *`mm/dd/yyyy`* format	
Hold	See also "Options" on page 484.	
	ISPF only) Required. Indicates whether to display only those events that have not been dispatched and currently have a HOLD status.	
	These are the valid values:	
	`Y`	Default. Display all non-dispatched events with a HOLD status.
	`N`	Do not display non-dispatched events with a HOLD status.
Hold	See "H (Resource Hold)" on page 457.	
Holidays	Optional. Specifies the date of each holiday your company observes, in *`mm/dd/yyyy`* format (the slashes are required). For holidays that occur on the same day every year, such as Fourth of July and Christmas, enter `****` for the year. For example, `07/04/****` and `12/25/****`. This way, you only have to define these holidays once. You can specify up to 30 holidays on a calendar.	
I/O	Indicates the dataset type. Required if the Dataset Name field is completed. These are the valid values:	
	`I`	Default. Input.
	`O`	Output.
Iccfci	See Chapter 7, "Generation Options" on page 519.	
Iccfcom	See Chapter 7, "Generation Options" on page 519.	
Iccflib	See Chapter 7, "Generation Options" on page 519.	
Iccfrec	See Chapter 7, "Generation Options" on page 519.	
Idcams	See Chapter 7, "Generation Options" on page 519.	
Iefu83	See Chapter 7, "Generation Options" on page 519.	
Igncat2	See Chapter 7, "Generation Options" on page 519.	

Field	Description
Initiator Id (System)	Specifies the initiator ID (*nnnn*). The abbreviations in the second column represent the days of the week, and under each day, you specify Y or N to indicate whether the initiator is available that day.
Initid	See "Initiator Id (System)" on page 459.
Internal Class	Used with external security to identify each resource type (up to eight alphanumeric characters long). For example, the internal class name for Event Master Records is Z$EMR. The internal class name is used for all references to that resource type in ESI documentation, messages, and commands. See the *ASG-Zeke Scheduling for z/OS User's Guide* for information on the different authority levels. **Note:** This field has no relation to the Class ID used in Zeke's own internal security. Below are listed the name, description, and authority levels for each internal class: Z$CATAL — Secures access to the Zeke database as a single unit. Resource name=CREATE/BACKUP/RESTORE/STATUS/BLOCK/CLEARCPU. Levels: READ enables Status and Backup functions; ALTER enables Create, Restore, and Block functions. Z$CLASS — Secures access to Zeke's own class records that are used in internal security. Default resource name format=(Zeke catalog-id). Levels: READ/UPDATE/ALTER Z$CMD — Secures access to Zeke operator commands. Default resource name format=(command verb). Levels: READ Z$CND — Secures access to calendar records. Default resource name format=(calendar-name). Levels: READ/UPDATE/ALTER Z$ECDR — Secures access to the ESI class definition records. Default resource name format=(Zeke catalog-id). Levels: READ/UPDATE/ALTER

Field	Description	
	Z$EMR	Secures access to the Event Master Records. Default resource name format=(userid.record-type). Levels: READ/UPDATE/ALTER
	Z$GOPT	Secures to the generation option records. Default resource name format=(Zeke catalog-id). Levels: READ/UPDATE/ALTER
	Z$NAME	Secures access to the customer name records. Default resource name format=(Zeke catalog-id). Levels: READ/UPDATE/ALTER
	Z$ONLINE	Secures access to the Zeke online functions. Default resource name format=(online menu option name). Levels: READ enables Read mode for that section of the online; UPDATE enables Write mode. Options to which you do not have access are not displayed on the Zeke Primary Menu. **Note:** Typically, this option is not necessary if all of the desired records have been secured.
	Z$OPER	Secures access to operator records used in Zeke's internal security. Default resource name format=(Zeke catalog-id). Levels: READ/UPDATE/ALTER
	Z$PINT	Secures access to the partition/initiator records. Default resource name format=(Zeke catalog-id). Levels: READ/UPDATE/ALTER
	Z$POOL	Secures access to the pool definition records. Default resource name format=(Zeke catalog-id). Levels: READ/UPDATE/ALTER
	Z$RESRC	Secures access to the resource definition records. Default resource name format=(Zeke catalog-id). Levels: READ/UPDATE/ALTER

Field	Description	
	Z$SCHED	Secures access to the SCHEDULE function. Default resource name format=(userid). The parameters provided by the user are used to build the resource name for one SAF call. Object-oriented security is not applied to this function because a security call would be required for each attempt to access a record. Since the SCHEDULE function involves reading every EMR in the Zeke database, calendars, and Zeke variables, and then creating SQRs, thousands of security calls could result, with a substantial impact on performance. Levels: If the ACTIVATE parameter is specified, ALTER authority is required. Otherwise, READ authority is required.
	Z$SET	Secures access to the ZEKESET functions. Default resource name format=(command verb). Levels: READ
	Z$SIM	Not valid at this time. In a future release, this class will secure access to the simulation function. Default resource name format=(userid). Levels: n/a
	Z$SQR	Secures access to the Schedule Queue Records. Default resource name format=(userid). Levels: READ/UPDATE/ALTER
	Z$VAR	Secures access to variables. Default resource name format=(userid). Levels: READ/UPDATE/ALTER
	Z$XCOM	Secures access to individual commands within an SCOM. Default resource name format=(command code). Levels: CONTROL
January - December (Special Calendar)	An asterisk (*) in the column indicates which day of the month is valid for scheduling jobs referencing this calendar. The days are listed across the screen and the months are listed down the left-hand column to create a matrix of dates. To deselect a date where an asterisk has been entered, replace the asterisk with a period (.).	

Field	Description
JCL (EMR)	The JCL source library and member information. *Batch Parameters:* All under the EVENT function — BIMLIBRARY, BIMMEMBER, CONDORMEMBER, CONDORPASS, CONDORVERSION, FNAME, FTYPE, JCLSTART, JCLSTOP, JESQJCL, LIBRARIAN, PANVALET, PDS, MEM, XXXXXXXX, ZEKEJCL
JCL (Schedule View)	ISPF only) An asterisk (*) indicates that online JCL exists for this event.
JCL (Zeke Online)	The JCL that resides in the Zeke database for this event. An unlimited number of lines is allowed, with up to 72 alphanumeric characters per line.
Jclcol71	See Chapter 7, "Generation Options" on page 519.
JCL1 - 5	See Chapter 7, "Generation Options" on page 519.
JES2 Job Id	Unique ID number (*nnnnn*) that JES2 assigns to the job. This ID is useful when you need to locate a particular job in the JES queue, especially if there are multiple jobs with the same name.
JES System ID /POWER ID	For z/OS systems, this is the JES system ID (up to eight alphanumeric characters long). This ID is required only if you are sharing a JES spool queue. A JES system ID is not required if the DispSel generation option is set to `N`: • If the GENSYS record does contain a JES system ID, then the ID is used in SYSAFF statement. • If the GENSYS record does not contain a JES system ID, then no SYSAFF statement is inserted in the JCL when Zeke dispatches the job. For VSE systems, this is the POWER ID (up to eight alphanumeric characters long). This ID is required only if you are sharing a POWER spool queue.
Job	The jobname as entered on the OS job card (up to eight alphanumeric characters long).
Job Execution Flow (Audit)	Required. Indicates whether you want to track and log all status changes an event goes through as it executes (event dispatched, event done, beginning of job, end of job, etc.). These are the valid values: `N` Default. Do not track and log an event's status changes. `Y` Track and log an event's status changes.
Job ID/Number	Displays the corresponding JES job ID.

Field	Description
Job Name	See "Job" on page 462.
Job ran __ Times	(Job events only) The number of job runs included in the job's current duration statistics. If this is less than the number indicated by the Durcount generation option, Zeke does not generate duration alerts or fail the job if it runs short or long. (Jobs are failed only if Durfail=Y or if the Fail option in the EMR is enabled.) This field is editable. If the job has not run enough times to satisfy the Durcount generation option, you can increase the **Job ran __ Times** value to start generating duration alerts/failures sooner. However, keep in mind that the Avgdur and Normal Range calculated from only a few runs may not be truly representative of the job's normal duration, and so the alerts/failures generated may not be appropriate. If the Avgdur field is updated, or if Clear Duration Stats is reset to Y, the **Job ran __ Times** field is reset to `1`.
Job step name	Optional. 8-character job step name. The auto reply is only valid if the message is issued in this step.
Jobfailc	See Chapter 7, "Generation Options" on page 519.
Jobs	See "Event Totals by Type" on page 443.
Jobxit#	See Chapter 7, "Generation Options" on page 519.
Last Accessed	See "Date Last Accessed/ Used" on page 436.
Last Backup	(Display) Date and time the database was last backed up.
Last Restore	(Display) Date the database was last restored.
Last Update (Event)	Displays the last modification date and time for the event record (and the user who modified the event).
Last Update (GENOPT)	(Display) Date and time the GENOPT was reloaded or was last updated and the user ID or batch jobname that made the update, or the user ID that reloaded the GENOPT (by issuing the ZRELOAD GENOPTS command). This information could also indicate whether the GENOPT was last reloaded automatically as result of a Zeke startup, or a database CREATE or RESTORE.
Last Used	See "Date Last Accessed/ Used" on page 436.
Late	See "Latestart (Schedule Time)" on page 464.

Field	Description
Late (SView Status)	ISPF only) Required. Indicates whether to display only those events that have not been dispatched and currently have a LATE status. If you do not specify a LATE time and the event is past its schedule time, Zeke does not consider the event late and does not issue LATE messages or display a LATE status. These are the valid values: `Y` Default. Display all non-dispatched events with a LATE status. `N` Do not display non-dispatched events with a LATE status.
Latealm	See Chapter 7, "Generation Options" on page 519.
Lateend (Schedule Time)	Specifies the time (in *`hh:mm`* format) by which the event must finish. The valid values range from `00:00` through `47:59`. If the Lateend time is reached but the event has not completed, or if the event is not *projected* to be completed by the Lateend time (i.e., if the event has not dispatched and the current system time plus the average duration is after the Lateend time), message Z0302I is issued to the console. **Note:** An event that is projected to finish late is not prevented from being dispatched unless its Mustend time is violated. Also, an event that is projected to finish late is not assigned a late status until the Lateend time is reached. If the time is greater than `24:00`, Zeke considers the event to require dispatching on the next day. **Note:** Late events can have a higher dispatch priority based on the generation option Prilate.
Latestart (Schedule Time)	The time by which the event must be dispatched, in *`hh:mm`* format. The valid values range from `00:00` through `47:59`. If the Latestart time is reached but the event has not dispatched yet, message Z0302I is issued to the console. An event that is projected to start late is not prevented from being dispatched unless its Notafter time is violated. Also, an event that is projected to start late is not assigned Late status until the Latestart time is reached. If the time is greater than `24:00`, Zeke considers the event to require dispatching on the next day.

Field	Description
	Note: Late events can have a higher dispatch priority based on the generation option Prilate. **Note:** If you use OpsCentral, an early warning alert is issued to OpsCentral if the event *is not projected* to be dispatched by the Latestart time (based on its predecessors),
LateStrt (Schedule View)	ISPF only) See "Latestart (Schedule Time)" on page 464.
Late End (Schedule View)	ISPF only) See "Lateend (Schedule Time)" on page 464.
Late Start (Schedule View)	ISPF only) See "Latestart (Schedule Time)" on page 464.
Late Time	See "Latestart (Schedule Time)" on page 464.
Length	Required on the ESI Customization - X1FRMAT1 screen and used with external security and SAF to specify the length (*nnn*) of a particular element when defining a resource name format. The resource name format determines what resource-related information is included in each security call. For example, if you defined a resource name to start in position 3 with a length of 5, then a security call for this resource would only search positions 3 through 5 for a match. See the *ASG-OASIS for z/OS Reference Guide* for additional information.
Level	Specifies the number of path levels to display when selecting events to add to the schedule based on a path. The valid values range from `1` through `999`. You can specify an asterisk (`*`) to display all levels. The default value is `1`.
Librblk	See Chapter 7, "Generation Options" on page 519.
Librdtf	See Chapter 7, "Generation Options" on page 519.
Librlub	See Chapter 7, "Generation Options" on page 519.
Librmgt	See Chapter 7, "Generation Options" on page 519.
Librmod	See Chapter 7, "Generation Options" on page 519.
Line	See "Note Information Lines" on page 473.

Field	Description
LoadComm	See Chapter 7, "Generation Options" on page 519.
Long Job Name	See "Full Job Name" on page 456.
Low	Required. Specifies the lowest condition code (up to four digits long) to be included in the range. This field is required if the condition code *range* (RA) operator is used; otherwise, leave this field blank.
LVL	Indicates the level of hierarchy in the job flow. `0` Original event or job as specified in the ZDISPLAY command. `1` Predecessor or successor of the specified event or job. `2` Predecessor or successor of a level 1 predecessor or successor.
Max *(Job Class Capacity)*	Optional. Job class capacity limit. If specified, Zeke submits jobs to JES according to this limit without considering whether an initiator is available. Because JES (along with Workload Manager, if used) assumes control over job execution, setting job capacity limits helps you retain more control over the priority. Controlling the number of Zeke-submitted jobs that are in the JES queue at the same time (instead of allowing Zeke to send an unlimited number) will increase the potential for the jobs to be executed in closer accordance with your Zeke-defined dispatching priorities. The valid values range from `0` through `999`. The default value is `NO` (i.e., no limit). **Note:** You must issue the ZRELOAD INIT command or cycle Zeke (ZKILL COLD or TRACK) for the changes to become effective.
MAXCOND AT EOJ	Optional. Indicates the highest condition code (*nn*) at the end of the job. Condition code checking is performed at EOJ for the maximum condition code from any step in this job (this is done independently of any step-level checking specified). You can specify operators, along with a single value or range of values. The only valid action that can be specified is `A`.
Max Share Count	See "Maximum Shared" on page 466.
Maximum Shared	Required. This number represents the quantity of a resource available to an event. The maximum value is `65535`.

Field	Description
Md (Resource Mode)	Indicates the resource mode required by the event: These are the valid values: `SR` (SHARE) Default. Multiple events have access to this resource. `EX` (EXCLUSIVE) Only one event has access to this resource. `ES` (EXCLUSIVE/SHARE) Only one event has access to this resource exclusively; however, other events can share the resource. Required if there is an entry in the Resource Name field.
Member	Optional. Specifies the PDS member name (up to eight alphanumeric characters long). If left blank, this field defaults to the jobname.
Message Text	The message text that Zeke searches for (up to 60 alphanumeric characters long). When a match is found, Zeke issues the appropriate auto reply.
Milestone	Indicates whether the event is a *milestone* event. A milestone event is a significant event in a job flow (in which events have predecessor/successor relationships) and must process on time to avoid a significant delay in the completion of the entire flow. The Milestone designation is valid for any type of event. These are the valid values: `N` Default. Event is not a milestone event. `Y` Event is a milestone event and is flagged in Schedule View and OpsCentral. Events flagged as milestones are not processed any differently than other events; this flag is simply a way to easily identify events considered to be milestones in a job flow. In Schedule View (ISPF only), milestone events are indicated by an asterisk (*) in the Milestone column. In OpsCentral, you can filter events that have been flagged as milestone events. You can also view (graphically) the positions of these events in a job flow. **Note:** If you use ASG-Unified Management Architecture, information about Zeke milestone events can be sent to UMA automatically. This field also can be used as selection criteria for selecting events.
Mode	See "Md (Resource Mode)" on page 467.

Field	Description
MON - SUN	(Display) Listed down each column are the dates for each day of the week during the requested month. To the right of each date is a code that indicates whether the event is scheduled. These are the valid values: `H` Holiday `W` Weekend (non workday) `S` Slack day (number of days between accounting calendars) `*` Event scheduled `*x` Event scheduled to occur *x* number of times that day
Monday Time Ranges - Sunday Time Ranges	Optional. For each day of the week, enter the start and stop times (in *hh:mm* format) that the initiator is available to Zeke. The valid values range from `00:00` through `24:00`. You can enter multiple time ranges for each day. The time ranges can also be modified by operator command. Examples: `00:00 - 24:00`—Available all day `08:00 - 20:00`—Available from 8 A.M. to 8 P.M. `00:00 - 00:00`—Unavailable for the entire day `00:00 - 07:00 08:00 - 24:00`—Available from midnight to 7 A.M. and again from 8 A.M. to 12:00 A.M. (midnight).
Month	Required. Specifies the month number that you want to view. The valid values range from `01` (January) through `12` (December). The default value is the current month.
Msgs	See "Event Totals by Type" on page 443.
Msg Text	See "Message Text" on page 467.
Msgalm	See Chapter 7, "Generation Options" on page 519.
Msgcod01 - 16	See Chapter 7, "Generation Options" on page 519.
Msgwait	See Chapter 7, "Generation Options" on page 519.
Mspintrl	See Chapter 7, "Generation Options" on page 519.

Field	Description
MTWTFSS	Displays the letters in this field represent the days of the week. One of these codes appears under each letter indicating whether the initiator has a specified time range for its availability to Zeke. These are the valid values: `Y` A time range is specified for that day and may be displayed on the System Initiator Detail screen. `N` No time range is specified; therefore, the initiator is available to Zeke all day from 00:00 to 24:00.
MultAp	See Chapter 7, "Generation Options" on page 519.
Multhit (EMR)	Required. Indicates whether an event is scheduled multiple times because of a nonworking day. For example, an event is normally scheduled on Mondays and Tuesdays; however, one week Monday is a holiday. So this field determines if the event is scheduled to run twice on Tuesday or if the Monday run is just skipped. These are the valid values: `Y` Default. Allow multiple schedule records. `N` Allow only one schedule record.
MultEn	See Chapter 7, "Generation Options" on page 519.
MultGr	See Chapter 7, "Generation Options" on page 519.
Multhit	See Chapter 7, "Generation Options" on page 519.
MultJn	See Chapter 7, "Generation Options" on page 519.
MultSys	See Chapter 7, "Generation Options" on page 519.
MultUs	See Chapter 7, "Generation Options" on page 519.
Mustend	Optional. Specifies the time (in *hh:mm* format) by which the event must be completed. Before an event is dispatched, Zeke adds the average duration and the current system time, and compares the result to this field. If the calculated time is greater than the 'must end' time, the event is no longer time-satisfied. Zeke removes the event from the dispatch queue and issues a message to the console. However, you can use the ZALTER MUSTEND command to force the job to run.
Must Time	See "Mustend" on page 469.
Name (Calendar)	See "Calendar Id" on page 431.

Field	Description
Name1-4 (Company)	Optional. Enter the company name you want displayed on all batch report headings on the first line. Enter any other information you want (such as mailing address) on lines 2 through 4. You can enter up to 40 alphanumeric characters per line. *Batch Parameters:* All under OPTIONS function—NAME1, NAME2, NAME3, NAME4.
Name (Event)	See "Ename" on page 441.
Name (Value Set By)	(Display) If the value was set by a user, this is the Zeke user ID. If the value was set by a program, this is the program name. If the value was set by a job, this is the jobname.
Name (Variable)	See "Variable Name" on page 509.
Name/Address (Audit)	Required. Indicates whether you want to track and log all changes to your company's name and address within Zeke. These are the valid values: `N` Default. Do not track and log name/address changes. `Y` Track and log all name/address changes.
NDSP	ISPF only) Required. Indicates whether to display events that have not been dispatched. These are the valid values: `Y` Default. Display all non-dispatched events. `N` Do not display non-dispatched events. **Note:** If you set NDSP to `Y`, Zeke displays all non-dispatched events (including those with a hold or late status, regardless of their setting). If you set NDSP to `N`, also set HOLD and/or LATE to `Y`, Zeke displays only non-dispatched events with a hold or late status.
Needs Operok	ISPF only) Required. Indicates whether to display events that are waiting for an operator okay. These are the valid values: `Y` Default. Display all events awaiting an operator OK.

Field	Description
	N Do not display events awaiting an operator OK.
Needs Timeok	ISPF only) Required. Indicates whether to display events whose time requirements have not been satisfied yet. These are the valid values: Y Default. Display all events that are not yet TIME satisfied. N Do not display events that are not yet TIME satisfied.
Needs Whenok	ISPF only) Required. Indicates whether to display events whose WHEN conditions have not been satisfied yet. These are the valid values: Y Default. Display all events that are not yet WHEN satisfied. N Do not display events that are not yet WHEN satisfied.
Netregid	See Chapter 7, “Generation Options” on page 519.
New Value (GENOPT)	The new generation option field value that has been updated and is pending reload.
New Value (Variables)	Optional. Specifies the new variable value. A character value can be up to 64 bytes long; a numeric value can range from -2,147,483,647 through +2,147,483,647.
nnn	ISPF only) Use to sort data on the Schedule View screen. Remove sort criteria by entering 0 (zero) in this field next to the item. Add sort criteria by entering any number and press Enter. This moves the field above the line to the *used* section. The value determines the field’s placement in the sort order. Order values begin at 10 and increment by 10. This allows you to position multiple fields in between existing fields, if necessary. Each time a field is included in or removed from the sort sequence, or fields are reordered, the Order values are recalculated for the entire list.
Nonwkday	See Chapter 7, “Generation Options” on page 519.

Field	Description
Normal Range	(This is displayed only for Job events.) The acceptable range of duration times for this event, in HH:MM:SS format. If duration alerts and/or duration failures are enabled for this event, executions that run shorter or longer than the Normal Range will generate an alert and/or be marked as failed. See the section on Event Activity Accounting in the *ASG-Zeke Scheduling for z/OS User's Guide* for details about how these values are calculated. This field can also be edited manually. **Note:** The high Normal Range value cannot be set to a lesser time than the Avgdur, and the low Normal Range value cannot be set to a greater time than the Avgdur. Since the Alert Tolerance, Normal Range, and Avgdur fields are interdependent, if you attempt to modify more than one field at once, the system will accept or ignore your changes based on this priority order: • Alert Tolerance • Normal Range (high) • Normal Range (low) • Avgdur For example, if you change all these fields, only the change to Alert Tolerance is accepted, and the system calculates the Normal Range values. If you change only the Normal Range low value, the system calculates the Normal Range high value and the Alert Tolerance; if you change only the Normal Range high value, the system calculates the Normal Range low value and the Alert Tolerance.
No. Tap	Optional. Specifies number of tape drives required by this job. The valid values range from `0` through `255`.
No. of Tapes Required	See "No. Tap" on page 472.
Notafter	Optional. Specifies the time after which an event cannot be dispatched, in *`hh:mm`* format. Before an event is dispatched, Zeke compares this field to the system time. If the NOTAFTER time is less than the system time, the event is no longer time-satisfied. It is removed from the dispatch queue and an operator message is issued. However, to force this job to run, use the ZALTER NOTAFTER command.
Note	An asterisk (*) indicates that a note documentation record exists for this event.

Field	Description
Note Information Lines	Optional. Enter any note information. You can add up to ten lines (up to 60 alphanumeric characters per line).
NUM	See "Automatic Reply Element Number" on page 429.
Number Dir Block	See "Variable Records" on page 509.
Number of Schedule Records in the Catalog	Displays the first line displays the number of non-work center events in the schedule. The second line displays the number of work center events in the schedule.
Number of Slack Days	Indicates the number (*nn*) of days between the end of this fiscal year and the start of the next one. This option is required if there are extra days between the end of this fiscal year and the start of the next one. If slack days are not specified as necessary, errors will occurs when the SCHEDULE function is executed.
Number of When Conditions for the SQRs	Displays the number of event segments that trigger other events. This number is not applicable for work center events, because work center events (WORK) do not use WHEN conditions.
Nwday (EMR)	Required. Code specifying how to schedule this event on nonworking days, such as holidays and weekends. This field specifies what action to take based on the keywords specified in the OCCURS clause for the event. These are the valid values: A — Default. Schedule the event after the nonworking day. B — Schedule the event before the nonworking day. O — Schedule the event on the nonworking day. N — Do not schedule the event. The Nonwkday generation option can be used to set this field automatically when a new EMR is added. **Note:** If you specify a holiday or weekend in the OCCURS clause, it overrides the Nwday field in the EMR.

Field	Description

OCCURS	The clause specifying the day or days that an event should be scheduled (up to 551 alphanumeric characters long). Each version of an event can have a different OCCURS clause.
	
	Note:
	On the Event Record Directory screen (ISPF), an asterisk (*) in the OCCURS field indicates that an OCCURS clause other than (REQUEST) has been defined for this event.
	
	The format of the OCCURS clause is:
	`OCCURS (keyword AND/OR keyword)`
	
	Note:
	When adding an OCCURS clause through the ZEKE batch utility, it is necessary to add the OCCURS parameter before the actual statement. It is also necessary to include parentheses surrounding the keyword(s). For example, `OCCURS (MONDAY.L)`
	
	These symbols are used in the OCCURS clause with keywords:
	
	`.` (period) — Separates two keywords. For example:
	`OCCURS (MONDAY.L)`
	This statement schedules the event on the last Monday of each month or period.
	`OCCURS (MONDAY.2)`
	This statement schedules the event on the second Monday of each month or period.
	
	`+` (plus) — Adds the specified number of days or workdays. For example:
	`OCCURS (MONDAY.L + 3 DAY)`
	This statement schedules the event three days after the last Monday of each month or period.
	`OCCURS (MONDAY.L + 3 WDAY)`
	This statement schedules the event three workdays after the last Monday of each month or period.

Field	Description

	- (minus) — Subtracts the specified number of days or workdays. For example:
	`OCCURS (TUESDAY.1 - 5 WDAY)`
	This statement schedules the event five workdays before the first Tuesday of each month or period.
	`OCCURS (DAY.L-2)`
	This statement schedules the event two days before the last day of each month or period.

	The OCCURS clause can contain one or more of these keywords:
	`DAILY` — Schedules the event on any day the Zeke SCHEDULE function runs, regardless of whether the day is a workday, weekend, or holiday.
	`WORKDAYS` — Schedules the event on any normal working day (not on holidays or weekends).
	`MONDAY` — Schedules the event on Mondays.
	`TUESDAY` — Schedules the event on Tuesdays.
	`WEDNESDAY` — Schedules the event on Wednesdays.
	`THURSDAY` — Schedules the event on Thursdays.
	`FRIDAY` — Schedules the event on Fridays.
	`SATURDAY` — Schedules the event on Saturdays.
	`SUNDAY` — Schedules the event on Sundays.
	`WMONDAY` — Schedules the event on working Mondays.
	`WTUESDAY` — Schedules the event on working Tuesdays.
	`WWEDNESDAY` — Schedules the event on working Wednesdays.
	`WTHURSDAY` — Schedules the event on working Thursdays.
	`WFRIDAY` — Schedules the event on working Fridays.
	`WSATURDAY` — Schedules the event on working Saturdays.
	`WSUNDAY` — Schedules the event on working Sundays.

Field	Description	
	JANUARY	Schedules the event in January. You can combine this keyword with other keywords. For example, this statement schedules an event on the last Friday in January: `OCCURS (FRIDAY.L AND JANUARY)`
	FEBRUARY	Schedules the event in February.
	MARCH	Schedules the event in March
	APRIL	Schedules the event in April.
	MAY	Schedules the event in May.
	JUNE	Schedules the event in June.
	JULY	Schedules the event in July.
	AUGUST	Schedules the event in August.
	SEPTEMBER	Schedules the event in September.
	OCTOBER	Schedules the event in October.
	NOVEMBER	Schedules the event in November.
	DECEMBER	Schedules the event in December.
	EOM	Schedules the event on the last day of the month.
	EOM-*xx*	Schedules *xx* days before the last day of the month.
	WEOM	Schedules the event on the last working day of the month.
	WEOM-*xx*	Schedules *xx* working days before the last working day of the month.

Field	Description	
	DAY *yy xx*	Schedules the event on the specified day of the month if the relationship is true, where: *yy* is an operator (i.e., LE, LT, GE, GT, NE, or EQ) or dot (.). *xx* is a number ranging from 01 through 31, or L, or L–. For example: OCCURS (DAY LE 07) This statement schedules the event on any day less than or equal to the seventh day of each month. OCCURS (DAY.5) This statement schedules the event on the fifth day of each month. OCCURS (DAY.L) This statement schedules the event on the last day of each month. OCCURS (DAY.L-1) This statement schedules the event on the day before the last day of each month.
	WDAY *xx yy*	Schedules the event on the specified workday of the month if the relationship is true, where: *xx* is an operator (i.e., LE, LT, GE, GT, NE, or EQ) or dot (.). *yy* is a number ranging from 01 through 31, or L, or L–. Examples: This statement schedules the event on the fourth workday of the month: OCCURS (WDAY EQ 04) This statement schedules the event on the fifth workday of each month: OCCURS (WDAY.5) This statement schedules the event on the last workday of each month: OCCURS (WDAY.L) This statement schedules the event on the workday before the last workday of each month: OCCURS (WDAY.L-1)

Field	Description	

	JDAY	Schedules the event on the specified day of the current year. JDAY uses the same format as the keyword DAY, but its period is for the year. JDAY can also be used to refer to the current Julian day in comparison phrases.
		Examples:
		This statement schedules the event on the first 31 days of the year:
		`OCCURS (JDAY LE 31)`
		This statement schedules the event on the last day of the year:
		`OCCURS (JDAY.L)`

	DATE *xx mm/dd/yyyy*	Schedules the event on the specified date if the relationship is true, where: *xx* is an operator (i.e., LE, LT, GE, GT, NE, or EQ). *mm/dd/yyyy* is an actual date. For example, this statement schedules the event on any date less than or equal to December 31, 2020: `OCCURS (DATE LE 12/31/2020)` **Note:** You can also use the date format *dd/mm/yyyy* if your OASIS SSSIOPT macro is set as DATE=DDMM.

	MONTH	Schedules the event during the specified month.
		Examples:
		This statement schedules the event on Monday of the eleventh month of the year (MONTH.11 is November in a regular calendar, but might be different if you are using a fiscal or user calendar):
		`OCCURS (MONDAY AND MONTH.11)`
		This statement schedules the event on the last fiscal month:
		`OCCURS (MONTH.L)`

	PERIOD	Schedules the event according to the specified period. Use this parameter with the AND parameter. For example, this statement schedules the event on the last Monday in the third period of the calendar: `OCCURS (MONDAY.L AND PERIOD.3)`

Field	Description	
	WEEK	Schedules the week in a month or period. For example, this statement schedules the event during the third week of every month or period: `OCCURS (WEEK.3)`
	FWEEK	(Full week) Schedules the event during a week in which all days of the week (Monday through Sunday) are in the same month or period. (It is irrelevant whether the week contains holidays.) For example, this statement schedules the event on Tuesday of the first complete week of the month or period: `OCCURS (TUESDAY AND FWEEK.1)`
	WDAYW	Schedules the working day of the week. For example, this statement schedules the event on the second workday of each week (which is based on the workdays and holidays specified in the calendar): `OCCURS (WDAYW.2)`
	WFWEEK	(Working full week) Schedules during a week in which all of the defined workdays (normally Monday through Friday) are in the same month or period. (It is irrelevant whether the week contains holidays.) For example, this statement schedules the event during the second week of the month or period that includes all normal workdays: `OCCURS (WFWEEK.2)`
	USEREXIT	Calls a user-written program during OCCURS clause processing. All calendar table information is passed to the exit. USEREXIT determines if the event should be scheduled. For example, this statement schedules if the user exit returns a response of true: `OCCURS (USEREXIT TESTOCCUR)` See the *ASG-Zeke Scheduling for z/OS Installation Guide* for more information on the User OCCURS Clause Exit.
	EVERY *xxx* DAYS BEGINNING *mm/dd/yyyy*	Schedules every *xxx* days beginning on the specified date. The event will not be scheduled prior to the specified date. **Note:** You can also use the date format *dd*/*mm*/*yyyy* if you have `DATE=DDMM` specified in your OASIS*xx* options member.

Field	Description	
	EVERY *xxx* WDAYS FROM *mm/dd/yyyy*	Schedules every *xxx* workdays prior to and after the specified date. When scheduling events for the future, the event will schedule prior to as well as after the specified date. **Note:** You can also use the date format *dd/mm/yyyy* if you have DATE=DDMM specified in your OASIS*xx* options member.
	REQUEST	Never automatically schedules this event. Add this event to the schedule with the ZADD command when a request to process is made.
	REFEVENT	Schedules according to the calendar (including nonworking days) and OCCURS clause of another event. If the OCCURS clause or calendar specification for the referenced event is changed, all events that reference the event is scheduled according to the changes. You can specify the event that you want to reference either by event number or event name. Because an event name might not be unique in the Zeke database, the first event that matches the event name is selected for the reference. **Caution!** Event numbers are unique; however, because Zeke assigns the event numbers automatically and can re-reassign available, previously-used numbers to new events, ASG recommends you reference other events by *event name* to avoid unintended references. This keyword can be used in combination with other OCCURS keywords. For example, this statement schedules the event two workdays after the OCCURS of event ASGJOB1: `OCCURS (REFEVENT ASGJOB1 + 2 WDAY)`

Field	Description	

	NOT	Schedules if the specified keyword condition is not true. This keyword can be used for a single condition or for a compound condition enclosed in parenthesis.
		Examples:
		This statement schedules the event on any day that is not defined to Zeke as a workday:
		`OCCURS (NOT WORKDAYS)`
		This statement schedules the event on all days except for Mondays and Tuesdays:
		`OCCURS (NOT (MON OR TUES))`
		This statement schedules the event on any working day that is not a Monday:
		`OCCURS (WORKDAYS AND NOT MONDAY)`

	AND	Schedules the event if both conditions are true.
	OR	Schedules the event if either of the conditions is true.
	Note: Use the keywords BEFORE, AFTER, and ON immediately following another OCCURS clause keyword or compound conditions enclosed in parenthesis to schedule the event when it would normally fall on a holiday or weekend. If one of these keywords is not specified, Zeke schedules the event the day after a holiday or weekend unless the NWDAY parameter specifies differently. See the *ASG-Zeke Scheduling for z/OS User's Guide* for more information on scheduling events on holidays and weekends.	
	BEFORE	Schedules the event on the working day prior to the normal selection day if the day is a holiday or weekend. (By default, events are scheduled on first working day after normal selection day.)
	AFTER	Default. Schedules the event on the working day after the normal selection day if the day is a holiday or weekend.
	ON	Schedules the event on the normal selection day even if normal selection day is a holiday or weekend.
	PERIOD	Schedules the event according to the specified period. For example, this statement schedules the event on the last Monday in the third period of the calendar: `OCCURS (MONDAY.L AND PERIOD.3)`

Field	Description	
	HOLIDAYS	Schedules the event on holidays. **Note:** To code an OCCURS clause with a relational reference to a holiday, such as `(HOLIDAYS + 1 DAY)`, the event master record's NWDAY parameter must be set to `O` (schedule the event on a nonworking day).
	WEEKENDS	Schedules the event on weekends.
	HOL/WEEK	Schedules the event on holidays and weekends.
	CALENDAR	Schedules based on the specified Special calendar. For example, this statement schedules the event only on the days marked in calendar TEST1: `OCCURS (CALENDAR TEST1)`
	VAR	Schedules the event if the specified Zeke variable is true. For example, this statement schedules the event only if the value for variable $SUE123 is `OKAY`: `OCCURS (VAR $SUE123 EQ OKAY)`
Occurs Date	OCCURS HIT date (in Julian format). When creating an event path, only events that would be scheduled on this date are included in the path. The default value is the current system date.	
Online Class Id	See "Class ID (Security)" on page 433.	
Online Functions	See "Allowed Functions" on page 427.	
Online Operator Id	See "Operator Id" on page 482.	
Op	See "Allowed Functions" on page 427.	
Operator Commands	Required. If you have set the associated user ID ZCOM to WRITE access, enter `Y` (to allow access) or `N` (to deny access) to the right of each Zeke operator command. The default value is `N`. As with the online functions, these commands can be limited further by the user ID on the Operator record. See Chapter 5, "Operator Commands," on page 253 for a complete list of valid schedule control (operator) commands.	
Operator Id	Specifies the user-assigned operator ID (up to eight alphanumeric characters long). The default ID is `OPERATOR` when the database is created.	

Field	Description
Operator (Condition Code Validation)	Required if the Stepname field contains a value. Specifies an operator that indicates when to perform the specified action: These are the valid values: `EQ` When the condition codes are equal. `LE` When the condition code is less than or equal to the specified condition code. `LT` When the condition code is less than the specified condition code. `GT` When the condition code is greater than the specified condition code. `GE` When the condition code is greater than or equal to the specified condition code. `NE` When the condition codes are not equal. `RA` When the condition code is in the specified range.
Oper Id	See "Operator Id" on page 482.
Oper Msg1 - Oper Msg6	The message text to issue to the system operator console when this message event is dispatched. Up to six lines are allowed, with up 60 alphanumeric characters per line.
Operok (SView Info Screen)	ISPF only) See "Needs Operok" on page 470.
Operok (EMR)	Required. Indicates whether Zeke is to wait for an operator OK before dispatching this event. These are the valid values: `N` Default. Do not wait for operator OK to dispatch. `Y` Wait for an operator OK before dispatching any event. **Note:** When a ZDISPLAY WAIT is issued, events that are waiting for an operator OK are shown as `AWAITING OPER OK`. Issue the ZOK command to run the events.
Operok	See Chapter 7, "Generation Options" on page 519.
Oprhold	See Chapter 7, "Generation Options" on page 519.

Field	Description

Opsys	The operating system type. These are the valid values:
	`z/OS`
	`VSE` (*use this value for VSE/SP*)
	`VSE/ESA`

Opt	See "Allowed Functions" on page 427.
Option	Name of the generation option field.
Options	See also "Allowed Functions" on page 427. Optional. Sets options for adding an event to the schedule manually. You specify `Y` or `N` for each option. Specify `Y` to perform the specified action when the event is added to the schedule. The default value for each option is `N`. These are the valid values: `AUTO` — Places an operator hold on the selected event and then re-create the schedule record. `ENABLE` — Re-activates a disabled schedule record and adds it back to the schedule. This action changes the event status from Disabled back to Enabled. `HOLD` — Place an operator hold on the schedule record. `REBUILD` — Re-create the SQR if one already exists. This option essentially: • Deletes the SQR • Re-adds the SQR • Resets all WHEN conditions • Reflects any EMR changes • Places the event into ACTIVE status • Resets any ZALTER changes previously made `REFRESH` — Refresh the schedule record by resetting the event as if it had not run. `RERUN` — Rerun the schedule record. `RUN` — Add the event to the schedule ready to run. This is the same as a ZALTER RUN operator command.
Order	ISPF only) Displays the order number for sequencing the sort order in the Schedule View Display Setup screen.

Field	Description
Owllib	See Chapter 7, “Generation Options” on page 519.
Owlmem	See Chapter 7, “Generation Options” on page 519.
Owlstat	See Chapter 7, “Generation Options” on page 519.
Pandisk	See Chapter 7, “Generation Options” on page 519.
Pandtf	See Chapter 7, “Generation Options” on page 519.
Pansys	See Chapter 7, “Generation Options” on page 519.
Parid	See “Partition Id (System)” on page 485.
Partid	See “Partition Id (System)” on page 485.
Part Id	See “Partition Id (System)” on page 485.
Part/Init	Displays the partition that set the variable.
Partition	Optional. Specifies the partition ID. The auto reply is only valid if the message is issued in this partition.
Partition Id (System)	Specifies the partition ID (up to four characters long). On the System Partition Specification screen, a partition ID is a one-character value for dynamic (D) partitions or a two-character value for static (S) partitions.
Partition/ Initiator (Audit)	Required. Indicates whether you want to track and log all changes to initiator and partition definition records. These are the valid values: `N` Default. Do not track and log partition changes. `Y` Track and log all partition changes.
Pauseeoj	See Chapter 7, “Generation Options” on page 519.
Pbtrack	See Chapter 7, “Generation Options” on page 519.
Pcom	VSE only. Optional. Enter any valid POWER commands (up to 60 alphanumeric characters long). See also “Event Totals by Type” on page 443.
Pctwarn	See Chapter 7, “Generation Options” on page 519.

Field	Description
PDS	Optional. Specifies the ddname of the partition dataset (up to eight alphanumeric characters long). Use the same ddname as specified in the Zeke started task procedure.
Pdsdd	See Chapter 7, "Generation Options" on page 519.
Pend	ISPF only) Required. Indicates whether to display events that have been dispatched with a pending status. These are the valid values: `Y` Default. Display all dispatched events with a PEND status. `N` Do not display dispatched events with a PEND status.
Pendinv	See Chapter 7, "Generation Options" on page 519.
Pendmsg	See Chapter 7, "Generation Options" on page 519.
Percent in Use	See "Event Records" on page 443.
Period 01 - Period 24	Optional. Specifies the number of days in each period (maximum is 40). You can define up to 24 periods.
Permanent	Required. Indicates whether the event is to remain in the schedule permanently/indefinitely. Permanent events are always available to respond to triggers, even during schedule load processing. See the Events chapter in the *ASG-Zeke Scheduling for z/OS User's Guide* for more information about permanent events. **Note:** In Schedule View (ISPF only), permanent events are indicated by an asterisk (*) in the Perm column. These are the valid values: `N` Default. Event is processed by each schedule run according to the OCCURS clause. `Y` Event must be added to the schedule using ZADD. After it has been added, the event remains in the schedule even across schedule runs until it is deleted using the ZDELETE command.

Field	Description
	Note: For permanent events, do not set a Times value (see page 504); permanent events can run an unlimited number of times. If a permanent event is later changed to be non-permanent, Zeke automatically sets the Times value to 1. Also, the Trig value (see page 504) for permanent events is always ALL. This field also can be used as selection criteria for selecting events.
Pjeclprt	See Chapter 7, "Generation Options" on page 519.
Pjob	VSE only. Optional. Specifies the VSE/POWER jobname (up to eight alphanumeric characters long). If left blank, Zeke uses the name entered in the Job field.

Platform	Required. Specifies the platform for the submitting system on which the JCL is executed. Zeke defaults to the platform of the system you are currently using. Valid entries are:
	
	`AIX` (see note below)
	`DCOSX` (Pyramid)
	`HPUX` (see note below)
	`MVS` (includes z/OS)
	`OS2`
	`OS400`
	`SUN` (see note below)
	`TANDEM`
	`USYS`
	`UNIX` (includes AIX, AT&T, HPUX, NCR, SCO, SunOS, Sun Solaris, etc.)
	`VMS`
	`VSE`
	`WINDOWS` (includes all supported versions)
	
	Note:
	Although the AIX, HPUX, and SUN platform codes listed above are supported, it is preferred that you use the UNIX platform code.
	
	Note:
	Zeke does not download jobs that have a platform of `MVS` or `VSE`.

Permanently save criteria	Required. Indicates whether to save the selection criteria as entered for the next time you access this screen. These are the valid values: `N` Default. Do not save selection criteria. `Y` Save selection criteria.

Field	Description
Pool (Audit)	Required. Indicates whether you want to track and log all changes to pool records. `N` Default. Do not track and log pool record changes. `Y` Track and log all pool record changes.
Pool Name	See "Pool ID" on page 488.
Pool ID	The pool ID (up to eight alphanumeric characters long). A *pool* is a group of one or more systems under one ID. If you have an event that is eligible to run under more than one system, you can specify a pool ID instead of a system ID in the SYS field on the Event Master Record.
Poolid Members	Displays the systems defined to the pool.
Posid	See Chapter 7, "Generation Options" on page 519.
Posidend	See Chapter 7, "Generation Options" on page 519.
POWER ID	VSE only. See "JES System ID /POWER ID" on page 462.
Prev Value	Displays the previous value of the variable.
Pri (Job)	Optional. Specifies a new priority code to replace the one on the job statement. The valid values range from range from `00` through `15`. If left blank, the priority code on the job statement is not changed.
Pri (REXX)	Specifies the OASIS ECF exec queue priority. The valid values range from `1` (highest) through `9` (lowest). The default value is `5`. Priority is used only if there is no free ECF task for the specified class when the event is dispatched. If so, the request is queued and this priority is used to determine which exec for the class executes when a task is available.
Prilate	See Chapter 7, "Generation Options" on page 519.
Process Option	Optional. Specifies how security access requests for a particular internal class are handled. These are the valid values: `N0` Do not call SAF. Grant all access requests. `N4` Do not call SAF. Use the application internal security (if applicable) to determine whether to grant the access request. If there is no internal security, access is allowed. `N8` Do not call SAF. Deny all access requests.

Field	Description	
	Y0	Call SAF. If SAF is unable to make a decision (the SAF return code is 4), grant the access request.
	Y4	Call SAF. If SAF is unable to make a decision (the SAF return code is 4), use the application internal security (if applicable) to determine whether to grant the access request. If there is no internal security, access is allowed.
	Y8	Call SAF. If SAF is unable to make a decision (the SAF return code is 4), deny the access request.
Proc Step	The name of the procedure step (up to eight alphanumeric characters long). Required if this step is within a procedure. An asterisk (*) can be used in any of the eight positions of the procedure step as a wildcard character to identify a generic step or generic procedure step name. For example, STEP5***.	
Procstep	See "Proc Step" on page 489.	
Program (Exec)	The program exec name (up to eight alphanumeric characters long). The auto reply is only valid if the message is issued by this program.	
Program	See "Program (Exec)" on page 489.	
Program Name	See "Program (Exec)" on page 489.	
Puser	VSE only. Optional. Specifies the user information (up to eight alphanumeric characters long); used for VSE/POWER accounting information.	
Range	See "High" on page 457. See "LVL" on page 466.	
Rdate	See "Work Date" on page 516.	
Rdrexit	See Chapter 7, "Generation Options" on page 519.	
Reason Code	See "Event Status/Reason Code" on page 443.	
REBUILD	See "Options" on page 484.	
REFRESH	See "Options" on page 484.	
Release Level	Displays the level of Zeke you are running.	
Removdq	See Chapter 7, "Generation Options" on page 519.	
Remtrig	See Chapter 7, "Generation Options" on page 519.	

Field	Description
Repcls	See Chapter 7, "Generation Options" on page 519.
RepJGrp	See Chapter 7, "Generation Options" on page 519.
RepJName	See Chapter 7, "Generation Options" on page 519.
RepJUser	See Chapter 7, "Generation Options" on page 519.
Reply	Optional. Specifies the reply you want Zeke to issue in response to the message (up to 60 alphanumeric characters long).
Reqopid	See Chapter 7, "Generation Options" on page 519.
RERUN	See "Options" on page 484.
Res	See "Allowed Functions" on page 427.
Resource	ISPF only) An asterisk (*) indicates that logical resources exist for this event.
Resource Definition (Audit)	Required. Indicates whether you want to track and log all changes to resource definition records. These are the valid values: `N` Default. Do not track and log resource definition changes. `Y` Track and log all resource definition changes.
Resource ID	See "Resource Name" on page 490.
Resource Name	Optional. Specifies the name of the resource (up to 44 alphanumeric characters long). Zeke checks the availability of these resources before the event is dispatched. **Note:** If the event's system ID is assigned to a pool, then each resource for that event should be set up as a global resource to ensure proper dispatching. *Batch Parameters:* RESNAME (EVENT function), GRESOURCE (OPTIONS function)
Restart	See "Allowed Functions" on page 427.
Restart and Rebuild All Conditions	ISPF only) Optional. Enter any character beside this field to refresh a job's schedule record by resetting all prerequisite completion indicators. All prerequisite conditions must then be set again. You can also refresh the schedule by entering the Zeke operator command ZREFRESH or ZALTER :

Field	Description
Restart and Satisfy All Conditions	ISPF only) Optional. Enter any character beside this field to satisfy these conditions: • TIMEOK • WHENOK • OPEROK • NOTDUROK • ZRELEASE These conditions are still checked before the job is dispatched: • SYSTEM HOLD • INITIATOR • TAPES • RESOURCES. You can also satisfy the conditions by entering the Zeke operator command ZREFRESH or ZALTER.
Retain (EMR)	Required. Indicates whether Zeke should retain the event if it is not dispatched on the scheduled date. These are the valid values: `Y` Default. Retain non-dispatched events during the next schedule run. `N` Discard non-dispatched events during the next schedule run.
Retain	See Chapter 7, “Generation Options” on page 519.
Retdays	See Chapter 7, “Generation Options” on page 519.
Retdone	See Chapter 7, “Generation Options” on page 519.
Retpend	See Chapter 7, “Generation Options” on page 519.
REXX exec	(z/OS only) Name of the member that contains the REXX exec. The dataset that contains this member must be defined in the SYSEXEC or SYSPROC DD concatenation of the OASIS ECF address space started task. The Zeke SZEKINS0 library contains the member REXSAMP, which is a sample REXX program that maintains control over the status (EOE or AEOE) of the event.
Route	Optional. Specifies the user-assigned route code that corresponds to the alternate console route code. The valid values range from `1` through `16`.

Field	Description
Rsrc	ISPF only) See "Resource" on page 490.
Run Date	Specifies the date (in *yyyyddd* format) on which the event is to run. Any events with this run date are selected.
Schd	See "Sched Time" on page 492.
Sched	See "Sched Time" on page 492. On Schedule Information Selection Criteria screen, the code that indicates whether to display events that are scheduled, but have not yet been dispatched. These are the valid values: `Y` Default. Display all dispatched events with a Scheduled status. `N` Do not display dispatched events with a Scheduled status.
Sched Date	See "Schedule Date" on page 492.
Sched Info	See "Scheduling Information" on page 493.
Sched Time	Specifies the normal schedule time (in *hh:mm* format). The valid values range from `00:00` (default) through `47:59`. If the time is greater than `24:00`, Zeke considers the event to require dispatching on the next day.
Schedule Control (Operator) Commands Allowed	Optional. Indicates whether each operator command (ZADD, ZALTER, etc.) is allowed for the online class ID. These are the valid values: `Y` The operator command is allowed for the class ID. `N` The operator command is not allowed for the class ID.
Schedule Date	The date the event was added to the schedule, in *yyyyddd* format. Any events with this schedule date are selected.
Schedule Queue Record (Audit)	Required. Indicates whether you want to track and log all changes to SQRs. These are the valid values: `N` Default. Do not track and log SQR changes. `Y` Track and log all SQR changes.

Field	Description
Scheduling Information	Displays the Julian date on which the event is scheduled to run.
Scom	See "Allowed Functions" on page 427. See "Event Totals by Type" on page 443.
SCOM Event Commands	See "Code (SCOM Event)" on page 434.
Scommax	See Chapter 7, "Generation Options" on page 519.
Scomwt	See Chapter 7, "Generation Options" on page 519.
Scp	ISPF only) Displays the scope of the resource definition. These are the valid values: `GBL` Defined to more than one Zeke system. `LCL` Defined to only one Zeke system.
Scratch	An asterisk (*) indicates that type of documentation record exists for this event.
Sdate	See "Calendar Start Date" on page 431. See "Work Date" on page 516.
Se	See "Allowed Functions" on page 427.
Sec	See "Allowed Functions" on page 427.
Secexitw	See Chapter 7, "Generation Options" on page 519.
Secfail	See Chapter 7, "Generation Options" on page 519.
Secgroup	Optional. For job events on z/OS, the security group (up to eight characters long). The SecUInit and SecULock generation options control whether this field is automatically populated and whether online users are permitted to update this field. The RepJGrp generation option controls whether Zeke replaces the GROUP= keyword on the JOB card with the Secgroup value when the job is submitted. For more details, see Chapter 7, "Generation Options" on page 519.
Sechide1	See Chapter 7, "Generation Options" on page 519.
Sechide2	See Chapter 7, "Generation Options" on page 519.

Field	Description
Secsel	See Chapter 7, “Generation Options” on page 519.
SecUInit	See Chapter 7, “Generation Options” on page 519.
SecULock	See Chapter 7, “Generation Options” on page 519.
Security	See “Allowed Functions” on page 427.
Security Class (Audit)	Required. Indicates whether you want to track and log all changes to internal security class records. These are the valid values: N — Default. Do not track and log internal security class changes. Y — Track and log all internal security class changes.
Security Operator (Audit)	Required. Indicates whether you want to track and log all changes to internal security operator records. These are the valid values: N — Default. Do not track and log internal security operator record changes. Y — Track and log internal security operator record changes.

Field	Description
Selection Criteria	Optional. You can perform a wildcard search on these types of fields: • Application • Calendar ID • DRL ID • Event name • Group ID • Jobname • Target • System ID • User ID There are selection criteria masks to search for events, calendars, and documentation records. An asterisk (*) in any position performs a wildcard search for all positions following it. For example, if you specify `ABC*` in the JOBNAME field, all events with jobnames that begin with ABC are selected. On the Schedule Information Selection Criteria screen, the Work Center Selection Criteria screen, and the Work Center Control Function - Load screen, a question mark (?) in any position performs a wildcard search for that one position. For example, if you specify `A?C`, all records that begin with A, end with C, and contain any character in the second position are selected.
Selection Field Masks	See "Selection Criteria" on page 495.
Set	(Display) Indicates whether a SET clause exists for the work center event. These are the valid values: `Y` A SET clause exists for the work center event. `N` A SET clause does not exist for the work center event.

Field	Description

Set (Clause)	The clause defining how the listed variables are to be set when the comment (work center) event is completed. The SET clause is used instead of WHEN conditions for work center events only, and can be up to 1360 alphanumeric characters long. One or more Zeke or OASIS variables are set to the specified values. For WORK events with multiple schedule records (versions), you can define a separate SET clause for each version of the event.
	
	Note:
	OASIS variables are available on z/OS only.
	
	Use these keywords in the SET clause statement:
	
	VAR
	XVAR — (VAR applies to Zeke variables; XVAR applies to OASIS variables.) The variable is automatically set to the specified value when the work center is flagged as complete. The operator does not have the option to change the value.
	
	For example, when the operator flags this work center as complete, the $JOB1 variable is automatically set to 45:
	
	`SET (VAR $JOB1 EQ 45)`
	
	?VAR
	?XVAR — (?VAR applies to Zeke variables; ?XVAR applies to OASIS variables.) The operator may enter the variable value before completion of the work center. However, the operator is not required to change the value. If OASIS variables are secured by SAF (through OASIS/ESI), the operator must have ALTER authority to OASIS variable value records to be able to set a variable value. See the *ASG-OASIS for z/OS Reference Guide* for more information on OASIS variables security.
	
	When the operator selects to complete an event with a ?VAR or ?XVAR, Zeke displays a screen to enter the variable value.
	
	For example:
	
	`SET (?XVAR JOB2 EQ 'MM/DD/YYYY')`
	
	`SET (?VAR $JOB2 EQ 'MM/DD/YYYY')`
	
	When the operator flags this work center as complete, the Work Center Control Functions - Page 2 screen is displayed. The operator is prompted to enter the current date.
	
	See the *ASG-Zeke Scheduling for z/OS User's Guide* for more information on setting variables with the SET clause.

Field	Description
Set Ver	See "Version" on page 509.
Shrd	Specifies the number (*nnnn*) of units of this resource that can be used in SHARE mode at the same time. The default value is `0001`. This value is required if the Resource Name contains a value.
Sta (Resource)	ISPF only) Displays the status of the resource. These are the valid values: `KPT` Resource is being held because while it was acquired and in use, an abend occurred. `ACQ` Resource is acquired by an event and is currently in use. `HLD` Resource is being held by an event until all of the required resources are available.
Start (EMR Accounting)	Displays the start date and start time of the last execution of the job during which the job ran and completed (SUCCESS, FAIL, F/SUCC, F/FAIL, or FAIL LONG). If the job has never dispatched, this field does not appear on the screen.
Start (Security)	Required. Used for external security to specify the starting position (*nnn*) for a particular element when defining a resource name format. The resource name format determines what resource-related information is included in each security call. See the *ASG-OASIS for z/OS Reference Guide* for additional information.
Start Date	See "Calendar Start Date" on page 431.
Starting Procname	Optional. Specifies the procedure name to start the rerun from (up to eight alphanumeric characters long).
Starting Stepname	Optional. Specifies the step name to start the rerun from (up to eight alphanumeric characters long).
Status	See "Download Status" on page 439.

Field	Description
Status (EMR Accounting)	Displays the status of the last execution of the job during which the job ran and completed. Values are: • SUCCESS (success, or success after failing once) • FAIL (failure) • F/SUCC (forced success, or forced success after failing once) • F/FAIL (forced failure) • FAIL LONG (marked as failed because its duration was too long) • FAIL SHORT (marked as failed because its duration was too short)
Status (Restart)	ISPF only) Optional. Indicates the restart status. These are the valid values: P Production R Restart
Status (Work Center)	(Display) Indicates whether the work center has been completed (DONE, DONE ONCE) or has not been completed (NOT DONE).
Status Time	ISPF only) Displays the time when the status of the event was last updated.
Status/Reason	See "Event Status/Reason Code" on page 443.
Step Name	Optional. Specifies the name of the job step that calls the cataloged procedure (up to eight alphanumeric characters long). An asterisk (*) can be used in any of the eight positions of the step name as a wildcard character. If a procedure is executed in this step, then the name of the step within that procedure should be specified either explicitly or with asterisks in the Procstep field on this screen. On the Event Resource Control screen, enter a step name. Zeke releases the resource after this step is completed. If the step is in a procedure, complete the Proc Step field.
Steplog	See Chapter 7, "Generation Options" on page 519.
Steplst	See Chapter 7, "Generation Options" on page 519.
Stepname	See "Step Name" on page 498.

Field	Description
Storage Needed for Loading SQRs to Memory	Displays the number of bytes needed to load the schedule records. **Note:** Work center (WORK) events are used to calculate this number only if the LoadComm generation option is set to Y (see page 554).
Storage Needed for the When Conditions	Displays the number of bytes needed to load the WHEN condition records.
Sttus Time	ISPF only) Displays the last time an event was updated with a particular status or reason code.
SubData	See Chapter 7, "Generation Options" on page 519.
Sys	See "System Id" on page 499.
System	See "System Id" on page 499.
Sysid	See "JES System ID /POWER ID" on page 462. See "System Id" on page 499.
System Id	The system ID to identify the system or pool (up to eight alphanumeric characters long). Depending on the screen, this ID can identify the system or pool that owns the event or variable. For an event, a field value *ANY indicates a pool of all systems in the Zekeplex and enables the event to be dispatched on any of the Zeke systems. First, the system ID must be defined in the NAME parameter of the OASIS*xx* options member. Then, you specify the system ID on the System Initiator/Partition Directory (GENSYS) screen. On the ISPF Audit Log Options screen, the default value is all asterisks ******** to indicate all systems. On the ISPF Resource Detail screen and the native System Resource Definition screen, this field is required if there is an entry in the Resource Name field. Enter the name of the system that owns the resource. The default value is GLOBAL (i.e., any system can share this resource). You can specify a resource name more than once with different system IDs. Any resource for an event whose system ID is assigned to a pool should be defined as a global resource to ensure proper dispatching.

Field	Description
	On the System Id Information screens for initiators, this field indicates the JES2 or system ID for a shared spool environment. If not using shared spooling, leave this field blank. On the Variable Record Functions screens, this field indicates the system that set the variable. On the Generation Options screen, this field indicates the Zeke system associated with the specified GENOPT. *Batch Parameters:* SYSTEMID (EVENT function), SYSGEN (OPTIONS function), GRESSYID (OPTIONS function)
System Id Members	The system IDs defined to the pool (up to eight alphanumeric characters each).
System Id Name	See "System Name" on page 500.
System Name	Required. System names are defined at installation time in the NAME parameter of the OASIS*xx* options member (up to eight alphanumeric characters long). On the System Id Information—Database Status Report screen, ******** indicates that statistics for all your defined Zeke systems are displayed.
System Type	See "Opsys" on page 484.
T/D	Code that identifies the dataset media. Required if the Dataset Name is completed. These are the valid values: `T` Tape. Default. `D` Disk
Tabular When Conditions	ISPF only) Optional. Specifies the WHEN condition and status of the prerequisites for the event. (Variable conditions are listed after all other conditions.) To satisfy individual conditions, tab to the blank space in front of the condition and enter any character. This action satisfies only that portion of the WHEN condition. These symbols in the space in front of the condition describe the status of the prerequisite: `*` Zeke satisfied the WHEN condition. `+` An operator command was used to manually satisfy the WHEN condition.

Field	Description
	# The WHEN condition is satisfied because the event is not in the schedule For information on how to satisfy all WHEN conditions, refer to the Schedule View Why screen. You can also satisfy the entire WHEN condition by entering the WOK or RUN line commands from the Schedule View screen. Variables cannot be set from this screen. **Note:** If there are existing WHEN conditions, you must add any additional WHEN conditions prior to satisfying the existing ones. Once you satisfy the WHEN conditions, Zeke will not recognize added WHEN conditions.
Tape	An asterisk (*) indicates that type of documentation record exists for this event.
Tapeio	See Chapter 7, "Generation Options" on page 519.
Tapes	Specifies the number of physical tape drivesrequired by the job. The valid values range from `0` through `255`. At event definition, use this field to indicate the estimated number of tape drives. If the Calctap generation option is set to `Y`, Zeke calculates this number based on the maximum number of drives ever used by this job and displays the value with an asterisk (*) to indicate that it is Zeke-calculated. If desired, you can enter a value in the Tapes field to override the calculated number. This information is stored on the system accounting record. The event is dispatched only when the specified number of drives is free. If it is time to dispatch and the drives are not free, Zeke informs the operator that the event is waiting on the necessary number of drives.
Target	Indicates the type of processing Zeke uses in determining where to send the JCL for execution. **Note:** Although the Target field appears on the Event Master Record functions screen for all events, it currently applies only to job events. It defaults to `*LOCAL` for all other event types. These are the valid values: `*LOCAL` Default. The job is dispatched, submitted and executed on the same system.

Field	Description	
	`*REMOTE` or `*R`	Remote processing. Zeke determines whether to use RL or RF processing based on the job's minimum required support.
	`*RMTLIM` or `*RL`	Remote limited processing. The job is dispatched and submitted on the same system, but uses NJE routing to execute on a different system with limited communication between the different systems (STORE and FORWARD). This type of processing does not support job cancellation due to condition code processing.
	`*RMTFUL` or `*RF`	Remote full processing. The job is dispatched and submitted on the same system, but uses NJE routing to execute on a different system with full two-way communication between the different systems (SEND WITH REPLY).
	other	Any valid 8-character Zeke Netregid. The job is dispatched to the remote system for submittal. **Note:** If the Netregid is not valid (i.e., if it is not found to be registered in the DMS network at the time of dispatch), Zeke does not dispatch the event. The event is placed on hold with a NET HOLD code.
Target Designation	See "Target" on page 501.	
TempJob	See Chapter 7, "Generation Options" on page 519.	
TempMsg	See Chapter 7, "Generation Options" on page 519.	
TempPcom	See Chapter 7, "Generation Options" on page 519.	
TempREXX	See Chapter 7, "Generation Options" on page 519.	
TempScom	See Chapter 7, "Generation Options" on page 519.	
TempVcom	See Chapter 7, "Generation Options" on page 519.	
TempWork	See Chapter 7, "Generation Options" on page 519.	
TempZcom	See Chapter 7, "Generation Options" on page 519.	

Field	Description
Template (Add Event)	The name of the template to use as a model when creating new events of this type (up to 12 alphanumeric characters long). To change one of these fields, simply type over the current name. In ISPF, once you change one of the template fields, the new name is displayed each time you access the screen until you change it again. In the native online system, these fields revert back to the names set in the generation options each time you start a new session. If you have multiple templates with the same event name and you enter that name in this field, Zeke will use the first template it finds with that name (the template with the lowest event number). To create a certain type of event, you must use a template of that type. For example, you cannot use a job event template to create a WORK event. **Note:** New events created from a template are created with the same event name as the template. ASG recommends that you change the event name to a different name.
Template (EMR)	Required. Indicates whether this event record is a model for other events of this type. These are the valid values: `N` Default. This event record is not a template. `Y` This event record is a template and can be used as a model for creating new events of this type. **Note:** Template events can never be scheduled like normal events—even if they are activated and have an OCCURS clause defined.
Template (Selection Criteria)	Optional. Indicates whether you want to list template events on the Event Record Directory screen. These are the valid values: `Y` List templates only. `N` List events only; do not list templates. (blank) Default. List events and templates.
Text	An asterisk (*) indicates that type of documentation record exists for this event.
Time (Current)	Displays the time this screen displayed.
Time (Variable)	Displays the time the variable was set to the particular value.

Field	Description
Timeok	ISPF only) See "Needs Timeok" on page 471.
Time Range	Optional. Format HHMMSS. In the first field, enter the starting time of the range you want to match. In the second field, enter the ending time of the range you want to match.
Times	Specifies the number of times this event is to be dispatched *per schedule run*. The valid values range from `1` through `9999`. If the number is greater than `1`, Zeke considers the event to be a *recurring* event. **Note:** For permanent events, do not set a Times value; permanent events can run an unlimited number of times. If a permanent event is later changed to be non-permanent, Zeke automatically sets the Times value to 1. **Note:** ASG recommends specifying a Freq time and/or a WHEN condition for permanent/recurring events.
Today	Displays the current date.
Total Var in Use	See "Variable Records" on page 509.
Tracking Control	See "Control" on page 435.
Trig	Applies to recurring events only. Indicates when the recurring event can satisfy WHEN conditions (i.e., serve as a trigger) for other events. A recurring event is one that has a Times value that is greater than `1` (i.e., an event that is dispatched more than once per schedule run). A non-recurring event is one that has a Times value of `1`; non-recurring events trigger regardless of the code specified in the Trig field. These are the valid values: A — Default. The recurring event can trigger other events *any* time it runs. **Note:** Permanent events (i.e., recurring events which can occur an unlimited number of times) always trigger on all occurrences. F — The recurring event can only trigger other events the *first* time it runs.

Field	Description
	L — The recurring event can only trigger other events the *last* time it runs. For example, suppose you have a recurring event that is scheduled to run every hour starting at 8:00 and ending at 12:00. If Trig is set to F, the event can satisfy WHEN conditions only on the 8:00 run, and all subsequent trigger calls for this event are ignored (until the event is rebuilt or refreshed). If Trig is set to L, the event can satisfy WHEN conditions only on the 12:00 run, and all other trigger calls for this event are ignored (until the event is rebuilt or refreshed). If Trig is set to A, the event can satisfy WHEN conditions on each run from 8:00 to 12:00.
Trigdt	See Chapter 7, "Generation Options" on page 519.
Trig Type	See "Trig" on page 504.
Trigger Name	Jobname or event that satisfies the WHEN condition for the jobname or event listed.
Trigger Type	See "Trig" on page 504.
Trigjob	See Chapter 7, "Generation Options" on page 519.
Trigrrn	See Chapter 7, "Generation Options" on page 519.
T-Ver	Version number for the triggering event.
Type (Calendar)	See "Calendar Type" on page 431.
Type (Event)	See "Event Type" on page 443.
Type (Partition)	VSE only. Required. Specifies the partition type. These are the valid values: D — Dynamic S — Static
Type (Path)	When selecting events to add to the schedule based on a path, this is the type of path to display. These are the valid values: P — Predecessors only. S — Successors only. B — Default. Both predecessors and successors.
Type (Variable)	Optional. Code that identifies the format of the variables you want to select. These are the valid values:

Field	Description
	C Character N Numeric
U83supp	See Chapter 7, "Generation Options" on page 519.
Unused	See "Event Records" on page 443.
Updated	Displays the date the documentation note information or JCL was last updated.
Usepjob	See Chapter 7, "Generation Options" on page 519.
Used	See "Event Records" on page 443.
User	See "User Id (Event)" on page 506. See "Userid" on page 506. See "User Id (Security)" on page 507. See "User Id (Variable)" on page 508. See "User Id (Work Center Selection)" on page 508.
Usercls	See Chapter 7, "Generation Options" on page 519.
User Id (Event)	A user-assigned code to identify the user ID for the event (up to eight alphanumeric characters long). The main function of the user ID is to secure the event from unauthorized users. This field also can be used to sort and select events by Schedule View (ISPF only), Report Writer, and Zeke operator commands. Zeke supports mixed-case user IDs; be sure to enter the desired user ID in the correct case (upper, lower, or mixed). The SecUInit and SecULock generation options control whether this field is automatically populated and whether online users are permitted to update this field. The RepJUser generation option controls whether Zeke replaces the USER= keyword on the JOB card with the Usrid value from the EMR. For more details, see Chapter 7, "Generation Options" on page 519.
Userid	See Chapter 7, "Generation Options" on page 519.

Field	Description
User Id (Security)	Optional. Specifies an event's user ID (up to eight alphanumeric characters long) either to limit or grant access to the event. You can enter a generic event user ID to either limit or grant access to a group of events. For example, if you enter PAY*****, you limit or grant access to all events with a user ID that begins with PAY. Zeke supports mixed-case security user IDs. In the native online system, use the PF10 toggle key to switch back and forth between CAPS N and CAPS Y modes while you are entering security user IDs. CAPS N mode accepts mixed-case user IDs (i.e., entry is case-sensitive); CAPS Y mode accepts upper case user IDs only (mixed-case or lower case entries are translated to all upper case). If you are currently making entries in CAPS N mode, the lower right portion of the screen displays the toggle option PF10=CAPS Y (i.e., pressing PF10 will toggle to CAPS Y mode). If you are currently in CAPS Y mode, PF10=CAPS N is displayed, and pressing PF10 will toggle to CAPS N mode.In ISPF, use the CAPS command to toggle between mixed-case and upper case modes. The current mode is displayed in the upper right portion of the Operator Detail screen. The authorized user IDs are checked in sequence. Therefore, it is possible to permit access to user ID BILL0501, but prohibit access to any other BILL**** user ID. Enter the BILL0501 user ID first in the list, permitting access. Then list the BILL**** user ID further down the list prohibiting access. Also, next to each user ID is a list of the Zeke record types that contain the Userid field. Enter whether access is denied or granted for this event for each type of record. Zeke also supports blank user IDs for operator records. Zeke allows a user ID mask to contain leading spaces, imbedded spaces, trailing spaces, or all spaces. **Caution!** When creating variables with a blank user ID in Zeke, the blank user ID must be set up to have Write security access to variables and work centers. To define a blank user ID on the Operator Detail screen (ISPF) or Online Operator Control screen (native), simply leave the Userid field blank. Type in the appropriate level of security access (W for Write, R for Read, etc.) for each category listed (Zcom, Events, Work Centers, Documentation, Variables) and press Enter.

Field	Description
User Id (Variable)	A user-assigned code (up to eight alphanumeric characters long) that identifies the user ID for the variable. The main function of the user ID is to secure the variable from being updated by unauthorized users. This field also can be used to sort and select variables by the Report Writer and Zeke operator commands. When creating variables with a blank user ID in Zeke, the blank user ID must be set up to have Write security access to variables and work centers.
User Id (Work Center Selection)	Optional. Specifies the user ID or any portion of the user ID you want to match and the appropriate wildcard characters (up to eight alphanumeric characters long).
Use Template	Required. Indicates whether you want to use a template when creating the new event. These are the valid values: `Y` Create the event from the template specified in the Template field. `N` Do not create the event from a template. **Note:** The value entered in the Use Template field is retained across ISPF sessions and within a ZEKEOL session.
Usrid	See "User Id (Event)" on page 506.
V	See "Allowed Functions" on page 427.
?Value	Displays the value to which to set the variable.
Value	Current value of the generation option field.
Value Set By	(Display) Indicates how the variable's value was set. These are the valid values: `J` Job `P` Program `U` User
Var	See "Allowed Functions" on page 427.
Variable	See "Allowed Functions" on page 427. See "Variable Name" on page 509.

Field	Description
Variable Name	Displays the Zeke variable name. All Zeke variables begin with a dollar sign ($).
Variable Records	Displays the number of Zeke variable (dataname) records in use and the number of blocks these records use.
Vcom	See also Allowed Functions on page 427 and "Event Totals by Type" on page 443. Required. Enter any VM command (up to 60 alphanumeric characters long), as long as the privilege class of the command and issuing machine coincide.
Ver (EMR)	See "Version" on page 509.
Ver (Dataset)	For input datasets, enter the version number. For example, enter `001` (default) for the most current, `002` for next most current, etc. Required if the Dataset Name field is completed.
Verload	Required. Specifies the number of versions of this event to be loaded during the schedule build. This field defaults to zero. If Verload is set to `0`, only one version of the event (version zero) can be in the schedule at a time. If Verload is set to one, only one version is created by the schedule build, but any number of versions can be added to the schedule after schedule load using the ZADD command (up to 32,767 versions). **Note:** ASG recommends running no more than 1,000 versions of a single event. Each version of an event operates independently of all other versions of the same event. For instance, you can add a JCL override for one version without affecting other versions of the same event. **Note:** Multiple versions of the same event all have the same jobname; therefore, they cannot execute concurrently.
Version	For events with multiple schedule queue records with the same schedule date (i.e., multiple versions), specifies the number (*nnnnn*) that identifies a particular version of the event. When adding an event from Schedule View or from an event path, this is the version of the event to be added to the schedule.
Version Number	See "Version" on page 509.
Versn	See "Version" on page 509.
Versn No.	See "Version" on page 509.

Field	Description
Viewing Documentation Note Information	See "Documentation Note Information" on page 439.
Viewing Documentation Scratch Pad	See "Documentation Scratch Pad Info" on page 439.
Viewing Documentation Text Information	See "Documentation Text Information" on page 439.
Virt Mem	See "Vmem" on page 510.
Virtual Memory Required	See "Vmem" on page 510.
Vmcmsin	See Chapter 7, "Generation Options" on page 519.
Vmem	VSE only. Specifies the amount (*nnnn*) of storage (virtual memory) required. If the CalcMem generation option is set to Y, this value is determined by Zeke, and is displayed with an asterisk (*) to indicate that it is Zeke-calculated.
Vmjclid	See Chapter 7, "Generation Options" on page 519.
Vmoperid	See Chapter 7, "Generation Options" on page 519.
Volldtf	See Chapter 7, "Generation Options" on page 519.
Vollub	See Chapter 7, "Generation Options" on page 519.
Vollxid	See Chapter 7, "Generation Options" on page 519.
Vols	(Display) For input datasets, the volume version number. For output datasets, the number of volumes required.
Volume Serial Numbers	(Display) For input datasets, the volume serial numbers. For output datasets, the word 'OUTPUT'.
W	See "Allowed Functions" on page 427.

Field	Description

WHEN	A clause specifying the conditions that must be satisfied before an event can be dispatched (up to 1360 alphanumeric characters long). If an event does not have a WHEN condition defined to it, then the event is dispatched according to the specified schedule time on the EMR. For events with multiple schedule records (versions) with the same schedule date, you can define a separate WHEN condition for each version of the event.
	
	Note:
	On the Event Record Directory screen (ISPF), an asterisk (*) in the WHEN field indicates that a WHEN condition has been defined for at least one version of this event.
	
	When adding a WHEN condition through the ZEKE batch utility, you must add the WHEN parameter before the actual statement. For example:
	`WHEN (AEOE JOBABC)`
	Use these conditions with the WHEN parameter:
	
	`AEOE` Abnormal End Of Event. Applies to any type of event, including work enters. For example:
	`WHEN (AEOE EVENT1)`
	
	`AEOJ` Abnormal End Of Job. For example:
	`WHEN (AEOJ JOB1)`
	
	`AEOP` Abnormal End Of Program. For example:
	`WHEN (AEOP PAYROLL)`
	
	`AEOS` Abnormal End of Step.
	For example, if the step that ended abnormally does not execute a procedure:
	`WHEN (AEOS STEP4..JOBX)`
	If the step that ended abnormally executes a procedure, the procedure name must be included:
	`WHEN (AEOS STEP2.PROCSTEP3.JOBX)`
	
	`AT` Jobs are dispatched only when the prerequisite(s) on the remote system are satisfied. AT is the Netregid of the remote system where the prerequisite is to occur. For example:
	`WHEN (EOJ JOBA AND EOJ JOBB AT SYSB)`
	A Netregid is limited to eight alphanumeric characters long See the *ASG-Zeke Scheduling for z/OS User's Guide* for more information on remote prerequisites.

Field	Description	
	BOJ	Beginning Of a Job. For example: `WHEN (BOJ JOB3)`
	BOP	Beginning Of a Program. For example: `WHEN (BOP PGMNME)`

	DSN	Close of an output z/OS dataset. The DSN WHEN condition is satisfied when a sequential, non-VSAM dataset is created and closed. For example,
		`WHEN (DSN DATASET.NAME)`
		`WHEN (DSN DATASET.NAME AND EOJ PAYROLL)`
		Before using DSN for non-VSAM datasets, ensure that the IEFU83 exit is active or that Connect:Direct product (formerly Network Data Mover, or NDM) is installed and active. If you do not have Connect:Direct installed and wish to use the DSN WHEN condition using a batch utility program, See the section on Connect:Direct support in the *ASG-Zeke Scheduling for z/OS Installation Guide* for more information.
		
		A generation data group (GDG) dataset name with a generation number of ’G0000V00' is satisfied when any dataset in its GDG is created and closed. For example:
		`WHEN (DSN A.B.G0000V00)`
		The condition above is satisfied when any generation in the GDG ’A.B' is created and closed.
		The IEFBR14 program does not close a dataset.
		Note:
		The Dscltrig generation option controls the dataset dispositions that qualify as triggers for DSN WHEN conditions.

	EOE	End Of an Event. Applies to any type of event, including Comment (Work Center) events. This keyword requires the event name. Do not use the event number. For example:
		`WHEN (EOE EVENT1)`
		`WHEN (EOE 'EVENT TWO')`
		`WHEN (EOE PAYROLL)`
		`WHEN (EOE 'EVENT PAYROLL')`

Field	Description	
	EOG	End of a group of events, including work center) events. This WHEN condition is satisfied when all events in the schedule that match the specified event name are in either SUCCESS or DISABLED status. In addition, at least one event must be in SUCCESS status. For example: `WHEN (EOG GRPPAY)` See the *ASG-Zeke Scheduling for z/OS User's Guide* for more information EOG WHEN conditions.

	EOJ	End Of Job. For example:
		`WHEN (EOJ JOBNAME1)`
		`WHEN (EOJ JOBNAME1 AND EOJ JOBNAME2 AND BOJ JOBNAME3)`
		`WHEN (EOJ JOB$(CYCLE))`

	EOP	End Of a Program. For example:
		`WHEN (EOP PROGRAM.JOB)`
		`WHEN (EOP PGMNAME1 OR EOP PGMNAME2)`

	EOS	Normal End of Step. The step name is the name of the job step that calls the cataloged procedure. The procedure step is the name of the procedure step. For example, when executing a procedure:
		`WHEN (EOS STEPNAME.PROCSTEP.JOBNAME)`
		When not executing a procedure:
		`WHEN (EOS STEPNAME..JOBNAME)`

	NOTDURING	Prevents jobs/programs from running at the same time. (Not applicable with JES3). For example:
		`WHEN (NOTDURING JOB JOBNAME)`
		`WHEN (NOTDURING PGM PGMNAME)`
		`WHEN (EOJ JOB1 NOTDURING JOB PRODCICS)`
		`WHEN (NOTDURING JOB *PAY NOTDURING PGM PAY**01A)`
		Note:
		NOTDURING conditions are not supported while the system is in SMF recording mode (initiated by the ZKILL TRACK command).
		See the *ASG-Zeke Scheduling for z/OS User's Guide* for more information on NOTDURING conditions.

Field	Description	
	SP	If you are using System Pooling, this keyword tells an event to run on the same Zeke system ID as its prerequisite event. For example: `WHEN (EOJ JOB1 SP)` The event does not run until JOB1 goes to normal EOJ, and it must run on the same system as JOB1. Therefore, when JOB1 goes to EOJ, Zeke changes the system ID of the schedule record for the event to the same system ID that JOB1 was processed on. See the *ASG-Zeke Scheduling for z/OS User's Guide* for more information on specifying the same system.
	VAR	Jobs are dispatched only when the variable is equal to the specified value. For example: `WHEN (VAR $ABC EQ YES)`

	VER	Jobs are dispatched only at the end of a specific version of the event. If the VER condition is omitted, it defaults to the same version SQR. For example, if JOBB version 3 is waiting on EOJ of JOBA but no version is specified, it will wait on EOJ of JOBA run under a version 3 SQR.
		`WHEN (EOE EVENT4 VER 88)`
		`WHEN (EOJ JOBC AND EOJ JOBC VER 2)`
		The VER condition can appear as part of any EOE, EOJ, EOP, EOS, or EOG clause (including their WEAK counterparts); however, it actually applies only to the SQR causing the trigger to be generated — not the job, program, or step.
		Use an asterisk (*) after the VER condition to trigger off of any version of the SQR. In this example, the event will trigger off of the next EOE on EVTB:
		`WHEN (EOE EVTB VER *)`
		Note:
		These conditions do not support use of the VER keyword
		`DSN`
		`VAR`
		`?VAR`
		`XEOE`
		`XEOJ`

Field	Description	
	WEOE	Weak End Of Event (soft). For example: `WHEN (WEOE EVT62)` See the *ASG-Zeke Scheduling for z/OS User's Guide* for more information on weak conditions.
	WEOG	Weak End Of Group (soft). All events can be disabled for the WEOG condition. WEOG is also satisfied if there are no matching events in the schedule. `WHEN (WEOG GRPA)` See the *ASG-Zeke Scheduling for z/OS User's Guide* for more information on weak conditions.
	WEOJ	Weak End Of Job (soft). For example: `WHEN (WEOJ JOBNAME)` See the *ASG-Zeke Scheduling for z/OS User's Guide* for more information on weak conditions.
	XEOE	Extended End of Event. For example, if this WHEN condition is defined for JOBC: `WHEN (XEOE EVTABC)` XEOE is satisfied if event EVTABC has run to SUCCESS or DISABLED status since the last time JOBC ran. XEOE supports the use of generic event names.
	XEOJ	Extended End of Job. For example, if this WHEN condition is defined for JOBC: `WHEN (XEOJ JOBA)` XEOJ is satisfied if JOBA has run since the last time JOBC ran. **Note:** The XEOE and XEOJ conditions do not support use of the VER or SP keywords.
When (Schedule View)	ISPF only) An asterisk (*) indicates that a WHEN condition has been defined for at least one version of this event.	
Whenok	ISPF only) See "Needs Whenok" on page 471.	
When Vr	ISPF only) See "Version" on page 509.	

Field	Description
Wktrgdn	See Chapter 7, "Generation Options" on page 519. See "Allowed Functions" on page 427.
Wo	See "Allowed Functions" on page 427.
Work	See "Allowed Functions" on page 427. See "Work Date" on page 516. See "Event Totals by Type" on page 443.
Work Center	See "Allowed Functions" on page 427.
Work Date	Displays the date the event was added to the schedule.
Workdays	Indicates whether each day of the week is a workday for scheduling. These are the valid values: `N` Default. Day is not a workday. `Y` Day is a workday.
X14Cnm1	Generation option. See "X14Cnm1" on page 604.
X14Cnm1L	Generation option. See "X14Cnm1L" on page 604.
X14Cnm2	Generation option. See "X14Cnm2" on page 604.
X14Cnm2L	Generation option. See "X14Cnm2L" on page 605.
X14Coml	Generation option. See "X14Coml" on page 605.
Year (Calendar)	Specifies the year (*yyyy*) during which the calendar is valid. Four asterisks (`****`) is valid only for a standard calendar and indicates that the calendar is valid for any year. Required on the Calendar Directory screen (ISPF) if a calendar ID is specified.
Year (Occurs)	Required. Specifies the year (*yyyy*) of the requested calendar.
Z	See "Allowed Functions" on page 427.
Zc	See "Allowed Functions" on page 427.
Zcom	See "Allowed Functions" on page 427. See "Event Totals by Type" on page 443.
Zcom (EMR)	Required. Specifies a Zeke operator command or combination of commands (up to 60 alphanumeric characters long).

Field	Description
Zeke Command (Audit)	Required. Indicates whether you want to track and log Zeke operator commands as they are issued. These are the valid values: N Default. Do not track and log Zeke operator commands. Y Track and log Zeke operator commands.
Zekectl	Generation option. See "Zekectl" on page 605.
Zeke Job Date	ISPF only) See "Work Date" on page 516.
Zeke Pgm Runs	(Display) Total number of Zeke program executions.
Zeke Special Calendar Id	See "Calendar Id" on page 431.
Zeke Standard Calendar Id	See "Calendar Id" on page 431.
Zekestep	VSE only. See "Zeke System" on page 517.
Zeke Step	See "Zeke System" on page 517.
Zeke Step Name	Optional. 8-character Zeke step name. The auto reply is only valid if the message is issued in this step.
Zeke System	See "System Id" on page 499.
Zeke User Calendar Id	See "Calendar Id" on page 431.
Zeke Variable (Audit)	Required. Indicates whether you want to track and log changes to Zeke variable records. These are the valid values: N Default. Do not track and log changes to Zeke variable records. Y Track and log changes to Zeke variable records.
ZPrdCom	Generation option. See "ZPrdCom" on page 606.
ZprdSEmr	Generation option. See "ZPrdSEmr" on page 606.

7 Generation Options

This chapter lists (in alphabetical order) all Zeke generation option fields (which are used to specify your Zeke operating criteria).

At installation, Zeke generation options are set to default values that are suitable for most data centers. ASG recommends that you review the options immediately after Zeke is installed and make any modifications. With gained experience using Zeke features, you might choose later to change these settings options to better meet your needs.

A collection of generation options is referred to as a *GENOPT table* or *GENOPT*. You can use GENOPTs to group together specific generation option settings that control a particular system or that you want to be used across multiple systems. These are the types of GENOPTs:

- Special GENOPT `*ACTIVE` contains all options currently loaded in memory and active (i.e., both global and local options) for the local Zeke system.
- Special GENOPT `*GLOBAL` contains options that require the same setting across all Zeke systems in the Zekeplex (i.e., that share the same database).
- Local GENOPTs contain options that control a local Zeke system.

Generation option fields are organized according to these predefined categories:

Category	Description
Audit	Audit options. These options control which types of activities are tracked and logged in the Zeke audit log file (or, optionally, in a data space log).
Dispatching	Options that affect event dispatching.
Exits	Options that affect Zeke and operating system exits.
General	All options that do not fall under another category.
JCL	Options related to JCL sources and processing.
Messages	Options that control the issuing of messages.
Scheduling	Options that affect event scheduling.
Security	Options that control security.

Category	Description
Traces	z/OS only. Data space trace options.
User interfaces	Options that affect the ISPF and online facilities.
Variables	Options related to Zeke variables.

Some generation options are activated immediately when they are saved, but most options require a reload for each system that is affected by the changes.

The generation options are reloaded automatically during each Zeke startup. Or, you can reload the generation options at any time by issuing this Zeke operator command:

```
ZRELOAD GENOPTS [FORCE]
```

For most generation options, the changes take effect immediately; however, some options do require a Zeke re-cycle (i.e., at least a ZKILL TRACK re-cycle and, in some cases, a ZKILL COLD). A ZKILL WARM re-cycle will not reload the generation options.

In this chapter, the description of each generation option field includes its corresponding batch parameter, functional category, option type (i.e., local or global), and reload requirement, as well as its valid values.

You can view and maintain the information stored in the GENOPTs and generation option fields through the Zeke online facility. Additionally, you can use the GENOPT function of the ZEKE batch utility program to maintain GENOPTs and generation option fields (see "GENOPTS—Maintain Generation Options" on page 68) and use the Report Writer (see "LIST GENOPTS—Zeke Generation Options" on page 239) to list them.

For information on how to access the Zeke generation option fields, and for procedures on common Zeke setup tasks that require you to update your Zeke generation options, see the *ASG-Zeke Scheduling for z/OS User's Guide*.

These are the Zeke generation options:

Option	Description
Abhold	(Abend hold) Category: Dispatching Type: Global Reload requirement: `ZRELOAD GENOPTS` Required. Indicates how to handle a recurring event abend. A recurring event is an event that is dispatched more than once per schedule run.

Option	Description
	These are the valid values: `N` Default. Refresh the event. `Y` Hold the event and issue a message to the operator. If this option is set to `Y` and one occurrence of a recurring event abends, then subsequent occurrences are held until the abended event obtains an EOJ status.
AddInact	(Add inactive events) Category: Scheduling Type: Global Reload requirement: `ZRELOAD GENOPTS` Required. Indicates whether to allow inactive events to be added via the ZADD command. These are the valid values: `Y` Default. Allow ZADD of inactive events. `N` Prevent ZADD of inactive events. Attempting to ZADD a deactivated event results in error message Z09DCW.
AuditAcF	(Audit Access Failure) Category: ZRELOAD GENOPTS Type: Global Reload requirement: `ZRELOAD GENOPTS` Required. Indicates whether to issue audit security failure messages to the system console. These are the valid values: Y Default. Issue security failure messages to the system console. N Do not issue security failure messages to the system console. These are the valid values: `N` Do not log audit security failure messages. `Y` Default. Log audit security failure messages in data space log, system log and JES log.

Option	Description
AuditCls	(Audit operator classes) z/OS only Category: Audit Type: Global Reload requirement: `ZKILL TRACK` Required. Code that specifies whether changes to internal security operator class records are tracked and logged (to the Zeke audit log file or data space log file). These are the valid values: `N` Default. Do not log changes to internal security operator classes. `Y` Log changes to internal security operator classes. Log/audit 'what' object type: `OPC` Data format: `OPC,`*`operclass`*
AuditCmd	(Audit commands) Category: Audit Type: Global Reload requirement: `ZKILL TRACK` Required. Code that specifies whether executed Zeke operator commands are tracked and logged (to the Zeke audit log file or data space log file). These are the valid values: `N` Default. Do not log executed Zeke operator commands. `Y` Log executed Zeke operator commands. Log/audit 'what' object type: `CMD` Data format: `CMD,`*`cmdname`*

Option	Description
AuditCnd	(Audit calendars) z/OS only Category: Audit Type: Global Reload requirement: `ZKILL TRACK` Required. Code that specifies whether changes to Zeke calendars are tracked and logged (to the Zeke audit log file or data space log file). These are the valid values: `N` Default. Do not log changes to Zeke calendars. `Y` Log changes to Zeke calendars. Log/audit 'what' object type: `CAL` Data format: `CAL,`*`calid`*`,`*`year`*
AuditEcd	(Audit external classes) z/OS only Category: Audit Type: Global Reload requirement: `ZKILL TRACK` Required. Code that specifies whether changes to external security class definition records are tracked and logged (to the Zeke audit log file or data space log file). These are the valid values: `N` Default. Do not log changes to ESI classes. `Y` Log changes to ESI classes.
AuditEmr	(Audit EMRs) z/OS only Category: Audit Type: Global Reload requirement: `ZKILL TRACK` Required. Code that specifies whether changes to Zeke EMRs are tracked and logged (to the Zeke audit log file or data space log file). These are the valid values: `N` Default. Do not log changes to EMRs.

Option	Description	
	Y	Log changes to EMRs. Log/audit 'what' object type: `EMR` Data format: `EMR,evtnum,jobname,ename,applid,grpid,userid`
AuditEvt	(Audit events) Category: Audit Type: Global Reload requirement: `ZKILL TRACK` Required. Code that specifies whether event execution flow (e.g., dispatch, start, end, etc.) is tracked and logged (to the Zeke audit log file or data space log file). These are the valid values:	
	N	Default. Do not log event execution flow.
	Y	Log event execution flow (e.g., dispatch, start, end, etc.).
AuditGop	(Audit GENOPTs) z/OS only Category: Audit Type: Global Reload requirement: `ZKILL TRACK` Required. Code that specifies whether changes to Zeke GENOPTs are tracked and logged (to the Zeke audit log file or data space log file). These are the valid values:	
	N	Default. Do not log changes to GENOPTs.
	Y	Log changes to GENOPTs. Log/audit 'what' object type: `GOP` Data format: `GOP,genoptname`

Option	Description
AuditNam	(Audit company information) z/OS only Category: Audit Type: Global Reload requirement: `ZKILL TRACK` Required. Code that specifies whether changes to your company name and address are tracked and logged (to the Zeke audit log file or data space log file). These are the valid values: `N` Default. Do not log changes to your company name and address. `Y` Log changes to your company name and address.
AuditOpr	(Audit operators) Category: Audit Type: Global Reload requirement: `ZKILL TRACK` Required. Code that specifies whether changes to internal security operator records are tracked and logged (to the Zeke audit log file or data space log file). These are the valid values: `N` Default. Do not log changes to operator records. `Y` Log changes to operator records. Log/audit 'what' object type: `OPI` Data format: `OPI,`*`operid`*
AuditPas	(Audit passwords) Category: Audit Type: Global Reload requirement: `ZKILL TRACK` Required. Code that specifies whether changes to Zeke license keys are tracked and logged. These are the valid values:

Option	Description	
	N	Default. Do not log changes to Zeke license keys.
	Y	Log changes to Zeke license keys.
AuditPin	(Audit initiators/partitions) z/OS only Category: Audit Type: Global Reload requirement: `ZKILL TRACK` Required. Code that specifies whether changes to initiator or partition definitions are tracked and logged (to the Zeke audit log file or data space log file). These are the valid values:	
	N	Default. Do not log changes to initiator or partition definitions.
	Y	Log changes to initiator or partition definitions. Log/audit 'what' object type: `PIT` Data format: `PIT,`*`pintname`*
AuditPoo	(Audit pools) z/OS only Category: Audit Type: Global Reload requirement: `ZKILL TRACK` Required. Code that specifies whether changes to pool records are tracked and logged (to the Zeke audit log file or data space log file). These are the valid values:	
	N	Default. Do not log changes to pools.
	Y	Log changes to pools. Log/audit 'what' object type: `POL` Data format: `POL,`*`poolname`*

Option	Description
AuditRes	(Audit resources) z/OS only Category: Audit Type: Global Reload requirement: `ZKILL TRACK` Required. Code that specifies whether changes to resource definition records are tracked and logged (to the Zeke audit log file or data space log file). These are the valid values: `N` Default. Do not log changes to resources. `Y` Log changes to resources. Log/audit 'what' object type: `RDS` Data format: `RSD,`*`resname`*`,`*`system`*
AuditSqr	(Audit SQRs) z/OS only Category: Audit Type: Global Reload requirement: `ZKILL TRACK` Required. Code that specifies whether changes to an SQR (including satisfied triggers) are tracked and logged (to the Zeke audit log file or data space log file). These are the valid values: `N` Default. Do not log changes to SQRs. `Y` Log changes to SQRs (e.g., updates made by an operator, a trigger in the WHEN segment is satisfied, etc.). Log/audit 'what' object type: `SQR` Data format: `SQR,`*`evtnum`*`,`*`Sdate`*`,`*`Rdate`*`,`*`ver`*`,`*`jobname`*`,`*`ename`*`,`*`applid`*`,`*`grpid`*`,`*`userid`*

Option	Description
AuditVar	(Audit variables) Category: Audit Type: Global Reload requirement: `ZKILL TRACK` Required. Code that specifies whether changes to variable records are tracked and logged (to the Zeke audit log file or data space log file). These are the valid values: `N` Default. Do not log changes to variables. `Y` Log changes to variables. Log/audit 'what' object type: `VAR` Data format: `VAR,`*`varname`*`,`*`applid`*`,`*`grpid`*`,`*`userid`*
Aur	(Auto replies) Category: Dispatching Type: Local Reload requirement: `ZKILL TRACK` Required. Indicates whether to enable the automatic response facility (i.e., auto replies). This facility is designed for job events that require message responses. These are the valid values: `N` Do not use the automatic response facility. `Y` Default. Use the automatic response facility (i.e., Zeke supports auto replies).
AurIntv	(Auto reply interval) Category: Dispatching Type: Local Reload requirement: `ZRELOAD GENOPTS` Optional. Specifies the interval (in seconds) to check automatic responses. The valid values range from `1` through `99`. The default value is `1`. A lower value tends to improve throughput in sites with many jobs that require automatic responses. A higher value tends to decrease system overhead in sites with fewer jobs requiring automatic response.

Option	Description
AurMsg	(Auto reply messages) Category: Messages Type: Local Reload requirement: `ZRELOAD GENOPTS` Required. Indicates whether to inform an operator of an automatic response. These are the valid values: `N` Do not issue a message. `Y` Default. Issue a message when an automatic response is supplied.
BatOprid	(Batch operator ID) VSE only Required. Default operator ID to use for security verification of batch functions (up to eight alphanumeric characters long). The default operator ID is `BATCH`. **Note:** Ensure that there is an operator ID record defined through internal security for this ID.
BatSec	(Batch security) VSE only Required. Indicates whether Zeke is to provide internal security for batch processing. These are the valid values: `N` Default. Do not check batch security. `Y` Check batch security.
BimAppl	(Bim-Edit application) Category: JCL Type: Local Reload requirement: `ZRELOAD GENOPTS` Optional. Name of the Bim-Edit application Zeke is to receive JCL from (up to eight alphanumeric characters long).

Option	Description
BimPasw	(Bim-Edit password) Category: JCL Type: Local Reload requirement: `ZRELOAD GENOPTS` Optional. Password Zeke passes to Bim-Edit to receive JCL (up to six alphanumeric characters long).
BimUid	(Bim-Edit user ID) Category: JCL Type: Local Reload requirement: `ZRELOAD GENOPTS` Required if using Bim-Edit. The user ID Zeke passes to Bim-Edit to receive JCL (up to four alphanumeric characters long). The default value is `OMIT`.
Bypjob	(Bypass POWER job statements) VSE only Required. Indicates whether Zeke is to bypass all POWER job statements other than the first. (Only `*LOCAL` jobs are affected.)
CalcMem	(Calculate memory) VSE only Required. Indicates whether to calculate virtual memory for events that do not have a VMEM specified on the EMR.
CalcTap	(Calculate tapes) Category: Dispatching Type: Local Reload requirement: `ZRELOAD GENOPTS` Required. Indicates whether to calculate tape drive usage automatically. **Note:** Zeke calculates the number tape drives required for a job based on the single step that uses the most tape mounts. For example, if a two-step job mounts three tapes in one step and two tapes in the next step, the calculated tape value is `3` since that is the maximum used by this job in any one step. Zeke counts all tape mounts on the same drive as requiring one tape drive. For tighter control, ASG recommends that you use logical resources. These are the valid values: `N` Zeke does not calculate a value for the Tapes field.

Option	Description	
	Y	Default. Zeke determines the maximum number of tape drives used by this job historically, and uses this number if the Tapes value is 0 or blank. The calculated value is displayed on the EMR screen with an asterisk indicating that it is a Zeke-calculated value. If the Tapes field is other than zero, Zeke does not calculate a value, even if the CalcTap value is Y. In other words, if you enter a value in the Tapes field, it will override the Zeke-calculated value (if applicable). If CalcTap is set to Y, Zeke verifies there are enough allocated tape drives before dispatching the event. Zeke recalculates at start of each dispatch cycle.
	Note: If the DispSel generation option is set to N, CalcTap is ignored (even if it is set to Y).	
ChgVal	(Changes to values) Category: Messages Type: Local Reload requirement: ZRELOAD GENOPTS Required. Indicates whether to display a message when a variable is updated. These are the valid values:	
	N	Do not display a message.
	Y	Default. Display message Z0402I when a variable value changes.
ChkSEnv	(Check scheduling environment) z/OS only Category: Dispatching Type: Global Reload requirement: ZRELOAD GENOPTS Required. This option controls whether Zeke considers the scheduling environment conditions before dispatching events (of any type). These are the valid values:	

Option	Description	
	N	Default. Zeke does not consider the scheduling environment before dispatching an event, regardless of whether the event has a SCHENV value specified in its SQR. If an event has a SCHENV value, the value is ignored.
	Y	Zeke considers the scheduling environment before dispatching an event. If a scheduled event has a SCHENV value specified in its SQR, Zeke dispatches the event only when the scheduling environment is active.
CmdCons	(Command routing) z/OS only Category: User interface Type: Local Reload requirement: `ZRELOAD GENOPTS` Required. Indicates where to route command responses. These are the valid values:	
	N	Route command responses to all consoles connected to the system.
	Y	Default. Only route command responses back to the console where the command was issued.
CmsFType	(CMS file type) Category: JCL Type: Global Reload requirement: `ZRELOAD GENOPTS` Required. Default CMS file type (up to eight alphanumeric characters long) for job events with JCL files that have a filename, but no file type specified in the EMR. The default value is `JCL`.	
CommCtl	(Work center control) Category: Scheduling Type: Global Reload requirement: `ZRELOAD GENOPTS` Required. Indicates whether to retain work center events in the schedule until the event is considered done or disabled. Operators can also choose to retain individual events on the EMR.	

Option	Description
	These are the valid values: `N` Discard work center events the next day regardless of completion status. **Note:** If the value for Retain in the EMR is `Y` and the value in the CommCtl option is `N`, the Commctl option overrides the definition on the EMR for work center events only. `Y` Default. Retain work center events until complete or disabled.
CondrDv	(Condor device) Category: JCL Type: Local Reload requirement: `ZRELOAD GENOPTS` Required if using Condor. Device type that Zeke will pass to Condor to receive JCL (up to six alphanumeric characters long). The default value is `SYS000`.
CondrLb	(Condor library) Category: JCL Type: Local Reload requirement: `ZRELOAD GENOPTS` Required if using Condor. Name of the library that Zeke will pass to Condor to receive JCL (up to four alphanumeric characters long). The default value is `OMIT`.
CondrVer	(Condor version) Category: JCL Type: Local Reload requirement: `ZRELOAD GENOPTS` (Required if using Condor) Specifies the Condor version number (*nnn*). The default value is `001`.
Datasub	(Variable substitution) VSE only Required. Indicates whether Zeke should substitute Zeke variables in VSE JCL statements before submitting the job to POWER.

Option	Description
DefDelOJ	(Delete override JCL default) Category: JCL Type: Global Reload requirement: `ZRELOAD GENOPTS` Required. Indicates the default setting for the Delete after next use option when override JCL is added to an SQR. These are the valid values: `N` — Default. Set the default value for the Delete after next use option to `N`. Override JCL added to the SQR is not deleted after the SQR is executed. `Y` — Set the default value for the Delete after next use option to `Y`. Override JCL added to the SQR is deleted after the SQR is executed.
DefDPrty	(Default dispatch priority) Category: User interface Type: Global Reload requirement: `ZRELOAD GENOPTS` Required. Default dispatch priority to be used if a dispatch priority is not specified on the EMR. The valid values range from `0` (highest) through `99` (lowest). The default value is `50`.
DefDspCl	(Default dispatch class) Category: Dispatching Type: Global Reload requirement: `ZRELOAD GENOPTS` Required only if the DispSel generation option is set to `N`. Single-character default job class to be used if the class is not specified on the EMR. The valid values range from `A` through `Z`, and `0` through `9`. See the section, "Initiator Processing" in Chapter 6, "Resources" in the *ASG-Zeke Scheduling for z/OS User's Guide* for more information.

Option	Description
DefJcl	(Default JCL source) Category: User interface Type: Global Reload requirement: `ZRELOAD GENOPTS` Required. Indicates the default JCL source if one is not specified on the EMR. The jobname is the default member name. These are the valid values: `BIM` Bim-Edit `CMS` CMS file `CONDOR` Condor `DRIVER` CA-Driver (VSE only) `ICCF` ICCF (VSE only) `JCLMAN` JCLMAN (VSE only) `LIBR` Librarian `OWL` OWL (VSE only) `PAN` Panvalet `PDS` Partitioned dataset member (z/OS only) `SLI` SLI (VSE only) `VOLLIE` CA Vollie (VSE only) `Zeke` ZEKE `Z14C` Source not supported by Zeke (JCL is supplied by the ZEKE14C user exit)

Option	Description
DefJPrty	(Default job priority) Category: User interface Type: Global Reload requirement: `ZRELOAD GENOPTS` Optional. Specifies the default job priority (*nn*) to be used when adding a new EMR. Or, specify `NO` if you do not want a default priority assigned to new EMRs. This option replaces the Defpprty option in earlier Zeke releases and supports all jobs on any platform. These are the valid ranges of job priorities (which vary depending on the operating system): AS/400 `01` through `99` VSE `00` through `09` z/OS `00` through `15` other `00` through `99`
DefOpid	(Default operator ID) Category: Security Type: Global Reload requirement: `ZRELOAD GENOPTS` Required if the ReqOpid generation option is set to `N`. Default operator ID (up to eight alphanumeric characters long) to be assigned for internal security when a matching operator security record cannot be found. The default value is `OPERATOR`. In VSE and CMS, this value is used when an undefined operator logs onto the online system. See "BatOprid" on page 529 for the VSE default for batch processing.

Option	Description

DefPltfm	(Default operating system)
	
	Category: User interface
	
	Type: Global
	
	Reload requirement: `ZRELOAD GENOPTS`
	
	Required. Operating system Zeke defaults to when an EMR is created (up to eight alphanumeric characters long): These are the valid platform codes:
	
	`AIX` (see note below)
	`DCOSX` (Pyramid)
	`HPUX` (see note below)
	`MVS` (includes z/OS)
	`OS2`
	`OS400`
	`SUN` (see note below)
	`TANDEM`
	`USYS`
	`UNIX` (includes AIX, AT&T, HPUX, NCR, SCO, SunOS, Sun Solaris, etc.)
	`VMS`
	`VSE`
	`WINDOWS` (includes all supported versions)
	
	Note:
	Although the AIX, HPUX, and SUN platform codes listed above are supported, it is preferred that you use the UNIX platform code.
	
	Note:
	Zeke does not download jobs that have a platform of `MVS` or `VSE`.

DefSysId	(Default system ID) Category: User interface Type: Global Reload requirement: `ZRELOAD GENOPTS` Required. Default system ID (up to eight alphanumeric characters long) for new events if a system ID is not specified on the EMR. You can specify an asterisk (`*`) to indicate that the system ID where the event is defined be used as the default. The default system ID is `A`. **Note:** This value should match the ID and/or NAME as specified in the OASIS*xx* options member.

Option	Description
DispDly	(Dispatch delay) Category: Dispatching Type: Local Reload requirement: `ZRELOAD GENOPTS` (Required if the DispSel generation option is set to `N`.) Number (up to three digits long) of seconds to wait between dispatches of pooled events. The default value is `030`. **Note:** The DispDly value is ignored if the Posid generation option is set to `Y` (see page 566).
DispSel	(Dispatch selection) Category: Dispatching Type: Local Reload requirement: `ZRELOAD GENOPTS` Required. Indicates whether Zeke selects the initiators. These are the valid values: `Y` — Default. Zeke selects the initiators. If you want to configure Zeke to select initiators generally (i.e., DispSel is set to `Y`), you also can define up to 36 single-character job classes to be treated as exceptions (as if DispSel is set to `N`). Job class exceptions can enable Zeke-controlled jobs to be submitted to JES without having to wait for an available initiator (even if initiators are managed by WLM). Exceptions also are useful if you simply want to limit the number of jobs that Zeke can submit for a particular class of initiators. See the chapter on resources in your *ASG-Zeke Scheduling for z/OS User's Guide* for more information. `N` — Zeke does not select the initiators. If using JES3, this option must be set to `N`.

Option	Description
	Consider these points if you set DispSel to `N`: • Initiator name does not appear in message Z0533I. • NOTDURING WHEN conditions are not supported; however, you can specify the start-up option `PLEXNOTD=YES` in your ZEKE*xx* PARMLIB option member to enable enhanced NOTDURING processing and support NOTDURING WHEN conditions. • Tape drive prerequisites specified in the EMR are ignored. • An autoreply defined for one partition is valid for all. • ZMAP command does not show initiator name. • ZDISPLAY AVAILABLE command is not valid. • INITIATOR parameter of the ZHOLD, ZRELEASE, ZALTER, and ZDISPLAY commands is not valid.
DropSel	(Drop selected events) Category: Scheduling Type: Global Reload requirement: `ZRELOAD GENOPTS` Required. Indicates whether to drop completed events from the current schedule when selection parameters are included with the SCHEDULE ACTIVATE command for a new schedule run. These are the valid values: `N` — The SCHEDULE function drops from the current schedule *all* completed events regardless of whether they match the selection parameters included with the new SCHEDULE ACTIVATE command. This option prevents completed events from accumulating with the creation of each new schedule. **Caution!** Incomplete events are discarded during each schedule run if they have Retain set to `N` in the EMR; therefore, if you run multiple SCHEDULE commands per day with selection parameters, this option might cause some incomplete events to be dropped unintentionally.

Option	Description	
	Y	Default. The SCHEDULE function drops from the schedule only completed events that match the selection parameters included with the SCHEDULE ACTIVATE command. **Caution!** This option enables completed events to remain in the schedule indefinitely (which might impact performance); therefore, ASG recommends that you review the SCHEDULE job output routinely for message Z08D8I (which indicates completed events retained in the database).
DsclTrig	(Dataset close triggering) z/OS only Category: Dispatching Type: Global Reload requirement: `ZKILL TRACK` Optional. Indicates the dataset dispositions that qualify as triggers for DSN WHEN conditions. These are the valid values:	
	`NA`	Default. Triggers when a NEW dataset is closed (even by an abending program).
	`N`	Triggers when a NEW dataset is closed. This value can be used alone or in combination with another code (e.g., NO, NM, or NOM.).
	`O`	Triggers when an OLD or SHR dataset is closed. Triggers when a NEW dataset is closed. This value can be used alone or in combination with another code (e.g., NO, OM, or NOM.).
	`M`	Triggers when a MOD dataset is closed. Triggers when a NEW dataset is closed. This value can be used alone or in combination with another code (e.g., NM, OM, or NOM.).

Option	Description	
	A	Triggers when a dataset with any of the above dispositions is closed by an abending program. This code can be added to any combination of the codes above. For example, setting DsclTrig to NOMA would trigger on the close of any of the dispositions. **Note:** If A is not specified, then any of the other disposition codes above *do not* trigger if the dataset is closed by an abending program.
	(blank)	Does not allow dataset close triggering.
DsnTrig	(DSN triggering) Category: Dispatching Type: Global Reload requirement: ZRELOAD GENOPTS Required. Indicates which SQRs to trigger when there are multiple SQRs in the schedule with the same event number and the same DSN trigger specified. These are the valid values:	
	TA	Default. Trigger all events waiting on the DSN trigger (all versions).
	ND	Trigger the event with the most recent schedule/run date (all versions).
	NT	If multiple events are waiting on the DSN trigger, do not trigger any events, and place them all on hold; else trigger SDATE.
	OD	Trigger the event with the oldest schedule/run date (all versions).
DSPBatch	(Data space use for batch processing) z/OS only Category: General Type: Local Reload requirement: ZRELOAD GENOPTS Indicates whether the ZEKE batch utility will use a dataspace when generating reports, creating the schedule, and backing up the database.	

Option	Description
	These are the valid values: `Y` Default. The ZEKE batch utility program will use a dataspace when generating reports, creating the schedule, and backing up the database. This is identical to specifying the DATASPACE parameter on the LIST, SCHEDULE, and BACKUP batch commands. Using a dataspace will improve performance. If a dataspace is not desired for a particular execution, the NODATASPACE parameter can be specified. `N` A dataspace is not used unless the DATASPACE parameter is specified on a batch command.
DSPIndex	(Data space index) z/OS only Category: General Type: Global Reload requirement: `ZKILL TRACK` Required. Code that specifies whether you want to create a data space to contain full index data for the events defined in the Zeke database. These are the valid values: `N` Do not create a data space; use common storage. `Y` Default. Build the full event index in a data space. The dataspace is named ZIX and is owned by the OASIS address space (which enables it to be maintained across a ZKILL WARM).. The maximum size of the data space is 2 GB. ASG recommends this setting for large databases for these reasons: • Improves ZADD command processing • Improves online browsing and retrieval of jobs • Enables the PATH, PREDECESSOR, and SUCCESSOR functions from the EMR **Note:** If your system is not eligible to use this option, you can use the option "EDBindex" on page 545 instead.

Option	Description
DSPSched	(Data space use for schedule load) z/OS only Category: General Type: Local Reload requirement: `ZRELOAD GENOPTS` Indicates whether Zeke will use a dataspace when loading the schedule (to reduce database I/O). These are the valid values: `Y` Default. Zeke estimates the size of the schedule in the database. If it estimates that I/O will be reduced, the database is read into a dataspace and the schedule is loaded or reloaded from there, rather than from the database. `N` Do not use a dataspace for the schedule load.
DurAlert	(Duration alerts) z/OS only Category: General Type: Global Reload requirement: `ZRELOAD GENOPTS` Required. Code that specifies whether to issue alerts when a job runs longer or shorter than the acceptable range of duration times. Alerts include console message Z0319W and OpCentral alerts (i.e., if the Zeke server for OpsCentral is active). These are the valid values: `N` Alerts is not issued; however, duration statistics still are updated in the EMR when the job ends. `Y` Default. Alerts are issued when jobs run long or short. Duration statistics are updated. **Note:** The Enable Duration Alerts field in the EMR enables you to override this option for a specific event.

Option	Description
DurCount	(Duration count) Category: General Type: Global Reload requirement: `ZRELOAD GENOPTS` Required. Minimum number (up to four digits long) of times a job must run before Zeke begins these actions: • Using the event's duration history to generate duration alerts • Failing jobs that run longer or shorter than the acceptable range of duration times. These jobs are failed only if DurFail is set to `Y` or if the Fail option in the EMR is enabled. The default value is `10`. **Caution!** If the number you specify is too small, Zeke might not collect sufficient history data to determine the job's normal duration range (and might report its short or long duration, in error).
DurFail	(Duration failure) Category: Dispatching Type: Global Reload requirement: `ZRELOAD GENOPTS` Required. Indicates whether Zeke should fail jobs that run longer or shorter than the acceptable range of duration times. Marking these jobs as failed prevents them from triggering successor jobs. These are the valid values: `N` — Default. Jobs that run longer or shorter than the acceptable range of duration times are not marked as failed for that reason. `Y` — When a job ends, if the duration was less than or greater than its normal range of duration times, the job is marked as failed, an AEOJ trigger is generated, and Zeke issues message Z8T02I. **Note:** This option does not cause a job to be cancelled. If a job abended or failed due to a condition code record, it is not failed because of its duration. A job is failed due to its duration only if it would have been marked successful otherwise.

Option	Description
	Note: The Fail if short or long field in the EMR enables you to override this option for a specific event.
DynSmf	(Dynamic SMF exits) z/OS only Category: Exits Type: Local Reload requirement: `ZKILL COLD` Required. Indicates whether Zeke dynamically installs the SMF exits. If you are using a subsystem other than SSSI, the SMF exits must be installed dynamically. These are the valid values: `Y` Default. Zeke dynamically installs the SMF exits. `N` You need to build the SMF exits.
EDBindex	(Event database index) z/OS only Category: General Type: Global Reload requirement: `ZKILL TRACK` Required. Code that specifies whether you want Zeke to build a condensed index of event information in common storage. enabling this option results in improved ZADD command processing. These are the valid values: `N` Do not build a condensed index. `Y` Default. Build a condensed index in common storage. This option requires approximately 20 bytes of common storage for every event in the database. **Note:** This setting is required to track externally submitted jobs (i.e., jobs whose JCL is contained in the JES job queue).

Option	Description
	Note: You can use the DSPIndex option to build a full event index in a dataspace (see page 542). If both options are set to Y, the DSPIndex option overrides EDBIndex.
EojWake	(Zeke wakes for EOJs) Category: Dispatching Type: Global Reload requirement: ZRELOAD GENOPTS Required. Indicates whether Zeke immediately checks for events to dispatch when any job reaches an EOJ (i.e., success) status. These are the valid values: Y — Default. Zeke checks the (local) dispatch queue at every EOJ. N — Zeke checks the (local) dispatch queue every 60 seconds.
ESIActv	(Activate ESI) z/OS only Category: Security Type: Global Reload requirement: ZRELOAD GENOPTS Required. Indicates whether to activate the OASIS External Security Interface (ESI). This does not necessarily mean that the Security Authorization Facility (SAF) is called. These are the valid values: Y — Activate ESI. The ESI component of OASIS is called by Zeke when security verification is required. N — Default. Do not activate ESI. Use Zeke's internal security, where applicable.

Option	Description
FairMod	(CA Librarian modification number) Category: JCL Type: Local Reload requirement: `ZRELOAD GENOPTS` Optional. CA Librarian modification number (up to six digits long). Valid only for Release 3.1 and earlier. **Note:** Contact your CA Librarian vendor if you do not know this number.
FairOpn	(CA Librarian options code) Category: JCL Type: Local Reload requirement: `ZRELOAD GENOPTS` Optional. CA Librarian options code (up to four digits long). Valid only for Release 3.1 and earlier. **Note:** Contact your CA Librarian vendor if you do not know this code.
FairRec	(CA Librarian record code) Category: JCL Type: Local Reload requirement: `ZRELOAD GENOPTS` Optional. CA Librarian record code (up to two digits long). Valid only for Release 3.1 and earlier. **Note:** Contact your CA Librarian vendor if you do not know this code.
Flush	(Flush VSE job) VSE only Required. Indicates whether VSE jobs are flushed when a job step abends.

Option	Description
GUIServ	(Zeke server) z/OS only Category: General Type: Global Reload requirement: `ZKILL TRACK` Required. Code that specifies whether to activate the Zeke server. These are the valid values: `N` Default. Do not activate the Zeke server. `Y` Activate the Zeke server.
IccfCI	(ICCF control interval) VSE only Category: JCL Type: Local Reload requirement: `ZRELOAD GENOPTS` Required. Specifies the ICCF control interval size (*nnnn*) for FBA devices. The default value is `2048`. To change this number, generate your own ICCF library.
IccfCom	(ICCF common library) VSE only Category: JCL Type: Local Reload requirement: `ZRELOAD GENOPTS` Required. Specifies the library number (*nnn*) of the ICCF common library, which is searched when an ICCF member is not found in the library number specified in the IccfLib option. The default value is `002`. Specify `0` if you do not want a common library search.
IccfLib	(ICCF events library) VSE only Category: JCL Type: Local Reload requirement: `ZRELOAD GENOPTS` Required. Specifies the number (*nnn*) of the default ICCF library for events that have ICCF members as the JCL source, but do not have a library specified on the EMR. The default value is `001`.

Option	Description
IccfRec	(ICCF records) VSE only Category: JCL Type: Local Reload requirement: `ZRELOAD GENOPTS` Required. Specifies the number (*nn*) of ICCF logical records per block if the ICCF library is located on a CKD DASD device. For regenerated ICCF systems, the value is `22` or the value of the NRECS parameter. This field is not required for FBA-resident ICCF libraries.
Idcams	(IDCAMS return code) VSE only Category: Dispatching Type: Local Reload requirement: `ZKILL COLD` Required. Indicates whether the IDCAMS return code is available.
Iefu83	(Install IEFU83 exit) z/OS only Category: Exits Type: Global Reload requirement: `ZKILL COLD` Required. Indicates whether to install the SMF exit IEFU83. This exit enables events to be triggered when a dataset is closed. These are the valid values: `N` Default. Do not install SMF IEFU83 exit. Zeke is not able to trigger events when a dataset is closed. `Y` Install SMF IEFU83 exit.
IgnCat2	(Ignore NOT-CAT2) z/OS only Category: Dispatching Type: Local Reload requirement: `ZRELOAD GENOPTS` Required. Indicates whether to ignore NOT-CAT2 messages, or to mark the job as AEOJ. These are the valid values: `N` Default. Pay attention to NOT-CAT2 messages and mark the job as AEOJ.

Option	Description	
	Y	Ignore NOT-CAT2 messages.
JclBrows	(JCL browse-only) z/OS only Category: User interface Type: Global Reload requirement: `ZRELOAD GENOPTS` Indicates whether to restrict non-Zeke JCL to browse-only mode on the Schedule View Zoom screen. These are the valid values:	
	N	Default. Enable any JCL accessible from Schedule View to be edited from the Zoom screen.
	Y	Enable non-Zeke JCL to be viewed (but not edited) from the Schedule View Zoom screen. Only Zeke JCL and override JCL can be edited from the Zoom screen.
Jclcol71	(JCL column scanning) Category: Variables Type: Global Reload requirement: `ZRELOAD GENOPTS` Required. Indicates whether to limit variable substitution to columns 1 through 71. These are the valid values:	
	N	Default. For VSE/CMS, scan columns 1 through 80; for z/OS, scan columns 1 through 240.
	Y	Variable substitution is performed only on columns 1 through 71. Variable names cannot end after column 71. Data in columns 72 through 80 are not shifted due to variable substitution.

Option	Description
JCL1 through 5	(JCL source EMR fields) Category: JCL Type: Local Reload requirement: `ZRELOAD GENOPTS` Optional. Indicates the JCL sources to display on the EMR screen as options. Enter one JCL source in each field. These are the valid values: `BIM` Bim-Edit `CMS` CMS file `CONDOR` Condor `DRIVER` CA-Driver (VSE only) `ICCF` ICCF (VSE only) `JCLMAN` JCLMAN (VSE only) `LIBR` Librarian `OWL` OWL (VSE only) `PAN` Panvalet `PDS` Partitioned dataset member (z/OS only) `SLI` SLI (VSE only) `VOLLIE` CA Vollie (VSE only) `Zeke` Zeke `Z14C` Source not supported by Zeke (JCL is supplied by the ZEKE14C user exit)
JobfailC	(Condition code job failures) z/OS only Category: Dispatching Type: Global Reload requirement: `ZRELOAD GENOPTS` Indicates how to handle jobs that abend due to a COND= parameter in the job card.

Option	Description
	These are the valid values: `N` Default. Zeke fails the job and satisfies AEOJ triggers. `Y` Zeke marks the job as successful and satisfies EOJ triggers.
Jobxit#	(Job exit table entry) VSE only Category: Exits Type: Local Reload requirement: `ZKILL TRACK` Optional. Number (*n*) that indicates which $JOBEXIT table entry to use as Zeke's $JOBEXIT entry. The default value is `0`.
LateAlm	(Late alarms) Category: Messages Type: Local Reload requirement: `ZRELOAD GENOPTS` Required. Indicates whether Zeke is to sound the console alarm when a late message is issued. These are the valid values: `Y` Sound an alarm with each late message. `N` Default. Do not sound an alarm with late messages.
LibrBlk	(CA Librarian block size) Category: JCL Type: Local Reload requirement: `ZRELOAD GENOPTS` Required. Specifies the CA Librarian block size (*nnnnn*). Zeke uses this number to determine the amount of storage required for Librarian subroutines. The default value is `12800`.

Option	Description
LibrDtf	(CA Librarian library file name) Category: JCL Type: Local Reload requirement: `ZRELOAD GENOPTS` Required. File name of the CA Librarian library Zeke uses to retrieve JCL (up to seven alphanumeric characters long). A DD statement with the same name must reference the Librarian library Zeke is to access. The default value is `OMITTED` (i.e., Librarian support is not required).
LibrLub	(CA Librarian library logical unit) VSE only Category: JCL Type: Local Reload requirement: `ZRELOAD GENOPTS` Required. Specifies the CA Librarian library logical unit (*nnn*). The default value is `000`.
LibrMgt	(CA Librarian management code) Category: JCL Type: Local Reload requirement: `ZRELOAD GENOPTS` Required. CA Librarian management code (up to four alphanumeric characters long), if Zeke must access PROD2 Librarian members. The default value is `NONE`.
LibrMod	(CA Librarian release) Category: JCL Type: Local Reload requirement: `ZRELOAD GENOPTS` Required. Indicates the CA Librarian release. These are the valid values: `Y` CA Librarian Release 3.1 or later. `N` Default. Any CA Librarian Release earlier than 3.1.

Option	Description
LoadComm	(Load work centers into SQT) Category: Scheduling Type: Global Reload requirement: `ZRELOAD GENOPTS` Optional. Indicates whether to load work center events into the schedule queue table (SQT) storage. These are the valid values: `Y` — Default. Load work center events into the schedule queue table (SQT) storage. This option loads work centers more efficiently, but requires above-the-line common storage. If the GUISrv option is set to `Y` (to enable access or complete work centers from an OpsCentral console or through Zeke Web Services), then you must have LoadComm set to `Y`. `N` — Do not load work center events into the schedule queue table (SQT) storage. **Note:** If this option is set to `Y`, then Zeke flags a work center event as late if the current time is greater than the event's 'late end' time, and the event completion process has not started (i.e., it is not yet in a pending or success status). If this option is set to `N`, the 'late end' time does not cause Zeke to flag a work center event as late.
Logall	(Log all messages) Category: Messages Type: Local Reload requirement: `ZRELOAD GENOPTS` Indicates whether to log all messages. These are the valid values: `N` — Default. Do not log all messages. `Y` — Log all messages.

Option	Description
Logerr	(Log error messages) Category: Messages Type: Local Reload requirement: `ZRELOAD GENOPTS` Indicates whether to log error messages. These are the valid values: `N` Do not log error messages. `Y` Default. Log error messages.
Logwarn	(Log warning messages) Category: Messages Type: Local Reload requirement: `ZRELOAD GENOPTS` Indicates whether to log warning messages. These are the valid values: `N` Do not log warning messages. `Y` Default. Log warning messages.
MaxCC	(Maximum condition code) z/OS only Category: Dispatching Type: Global Reload requirement: `ZRELOAD GENOPTS` Required. Value ranging from `0` through `4095` that specifies the default maximum job condition code for job events that do not have condition codes specified. The default value is `NONE`, which indicates not to set a maximum condition code. If a Zeke-controlled job ends with a maximum condition code greater than the MaxCC value, the event is marked as failed.

Option	Description
MsgAlm	(Message event alarm) Category: Messages Type: Local Reload requirement: `ZRELOAD GENOPTS` Required. Indicates whether Zeke is to sound the console alarm when a message event runs. These are the valid values: `Y` Sound the alarm when a message event runs. `N` Default. Do not sound the alarm.
MsgCod*n*	(Message routing code) z/OS only Category: Messages Type: Local Reload requirement: `ZRELOAD GENOPTS` Required. Indicates whether to route messages to the specified message routing code (0 through 16). Zeke uses this code when issuing a WTO or WTOR. These are the valid values: `N` Default. Route messages to the specified routing code (0 through 16). `Y` Do not route message to the specified routing code.

Option	Description

MsgWait	(Message wait interval)
	Category: Messages
	Type: Local
	Reload requirement: `ZRELOAD GENOPTS`
	Required. Amount of time Zeke waits between dispatches of certain messages that indicate an event is waiting in the dispatch queue. The default is one hour. Use the format *`hh:mm`*.
	This option affects these messages:
	`Z0601I`
	`Z0602I`
	`Z0611I`
	`Z0615I`
	`Z0617I`
	`Z0618I`
	`Z0628I`
	`Z0631I`
	`Z0634I`

MspIntrl	(Must-start interval) Category: Dispatching Type: Global Reload requirement: `ZRELOAD GENOPTS` Required. Enter a time interval (in *`hh:mm`* format) that will determine when an event with a 'must start' or 'not after' time is considered at risk. An event is assigned a higher dispatch priority if its 'must start' or 'not after' time is within the specified interval. The default value is `01:00` (i.e., one hour). For example, if an event's 'must start' time is 3:00 P.M. and the interval time is one hour, then at 2:00 P.M. the event is assigned a higher dispatch priority. Zeke calculates the 'must start' time by subtracting the average duration from the user-defined 'must end' time.
MultHit	(Multiple schedule hits) Category: Scheduling Type: Global Reload requirement: `ZRELOAD GENOPTS` Required. Indicates whether an event is scheduled multiple times due to a nonworking day. If not specified on the EMR, the default value is `Y`. These are the valid values: `Y` Default. Allow multiple schedule records. `N` Allow only one schedule record.

Option	Description
MultAp	(Multiple events by application ID) Category: Scheduling Type: Global Reload requirement: `ZRELOAD GENOPTS` Required. Indicates what to do when a ZADD by application ID is issued and multiple events have the same application ID. This option is only in effect if multiple matching events are found. If only one event is found, it is added to the schedule. These are the valid values: `F` Default. Add the first matching event and end the search; this is the most efficient. `N` Do not add any events; list all matching events on the console. `A` Add all the matching events.
MultEn	(Multiple events by event name) Category: Scheduling Type: Global Reload requirement: `ZRELOAD GENOPTS` Required. Specifies what to do when a ZADD by event name is issued and multiple events have the same event name. This option is only in effect if multiple matching events are found. If only one event is found, it is added to the schedule. These are the valid values: `F` Default. Add the first matching event and end the search; this is the most efficient. `N` Do not add any events; list all matching events on the console. `A` Add all the matching events.

Option	Description
MultGrp	(Multiple events by group ID) Category: Scheduling Type: Global Reload requirement: `ZRELOAD GENOPTS` Required. Indicates what to do when a ZADD by group ID is issued and multiple events have the same group ID. This option is only in effect if multiple matching events are found. If only one event is found, it is added to the schedule. These are the valid values: `F` Default. Add the first matching event and end the search; this is the most efficient. `N` Do not add any events; list all matching events on the console. `A` Add all the matching events.
MultJn	(Multiple events by jobname) Category: Scheduling Type: Global Reload requirement: `ZRELOAD GENOPTS` Required. Indicates what to do when a ZADD is issued by jobname and more than one event has the same jobname. (The ZADD operator command adds the specified job to the schedule.) This option is only in effect if multiple matching events are found. If only one event is found, it is added to the schedule. These are the valid values: `F` Default. Add the first matching event and end the search; this is the most efficient. `N` Do not add any events; list the matching event numbers on the console. `A` Add all the matching event names.

Option	Description
MultUs	(Multiple events by user ID) Category: Scheduling Type: Global Reload requirement: `ZRELOAD GENOPTS` Required. Specifies what to do when a ZADD by user ID is issued and multiple events have the same user ID. This option is only in effect if multiple matching events are found. If only one event is found, it is added to the schedule. These are the valid values: `F` Default. Add the first matching event and end the search; this is the most efficient. `N` Do not add any events; list all matching events on the console. This is the most restrictive setting. `A` Add all the matching events. This is the least restrictive setting.
MultSys	(Multiple Zeke systems) Category: General Type: Global Reload requirement: `ZKILL TRACK` Required. Code that specifies whether the Zeke database is shared by more than one Zeke system (real or virtual). **Note:** The Zeke database cannot be shared by Zeke systems at varying release levels. These are the valid values: `N` The Zeke database is not shared. If you are running Zeke in a single system environment, ASG recommends you set MultSys to `N` to reduce processing overhead. `Y` Default. The Zeke database is shared.

Option	Description
Netregid	(System network registration ID) Category: General Type: Local Reload requirement: `ZKILL TRACK` Optional. Enter Zeke's logical network registration ID, used to identify the Zeke system in the OASIS/DMS network (up to eight alphanumeric characters long). This name must be unique across all products and platforms within the network, that is, your Zeke Netregid cannot be the same as any other product's Netregid on any platform. This Netregid should be used in the Target field on the Event Master Record Functions screen for any remote jobs that are to execute on this system. The Netregid can be any combination of characters; however, it cannot be a single period (.), and it cannot start with an asterisk (*).
Nonwkday	(Nonworking day scheduling) Category: Scheduling Type: Global Reload requirement: `ZRELOAD GENOPTS` Required. Indicates when to schedule an event if the OCCURS clause is true for a nonworking day. Zeke uses this code as the default when adding an EMR. These are the valid values: `A` Default. Schedule the event after the nonworking day. `B` Schedule the event before the nonworking day. `O` Schedule the event on the nonworking day. `N` Do not schedule the event.
OperOk	(Operator approval) Category: Dispatching Type: Global Reload requirement: `ZRELOAD GENOPTS` Required. Indicates whether Zeke is to wait for an operator OK (ZOK command) before dispatching any event. These are the valid values:

Option	Description	
	Y	Wait for an operator OK before dispatching any event. This value overrides the value of the Operok field in the EMR.
	N	Default. Do not wait for an operator OK to dispatch events. Dispatch the event when it is ready (unless an operator OK requirement was specified in the EMR).
OprHold	(System hold on startup) Category: General Type: Local Reload requirement: `ZKILL TRACK` Required. Indicates whether to place the Zeke system on hold upon startup. These are the valid values:	
	Y	Place the Zeke system on hold upon start up. You must issue the ZRELEASE SYSTEM command to release the Zeke system hold before Zeke can dispatch any events.
	N	Default. The Zeke system is ready to dispatch events upon startup.
OwlLib	(OWL LIBSPACE) VSE only Category: JCL Type: Local Reload requirement: `ZRELOAD GENOPTS` (Required if using OWL.) Specifies the amount (*nnn*) of memory Zeke needs to obtain the LIBSPACE work area of the OWL access method. Refer to your *OWL System Programmer Guide* for the formula to determine this value.	
OwlMem	(OWL MEMSPACE) VSE only Category: JCL Type: Local Reload requirement: `ZRELOAD GENOPTS` Optional. Specifies the amount (*nnnn*) of memory Zeke needs to obtain the MEMSPACE work area of the OWL access method. Refer to your *OWL System Programmer Guide* for the formula to determine this value.	

Option	Description
OwlStat	(OWL status code) VSE only Category: JCL Type: Local Reload requirement: `ZRELOAD GENOPTS` Required if using OWL. Default status code to use for an event that has an OWL member defined as the JCL source, but is missing an OWL status (up to four alphanumeric characters long). The default value is `PROD`.

PanDisk	(CA Panvalet library DASD)
	Category: JCL
	Type: Local
	Reload requirement: `ZRELOAD GENOPTS`
	Required. Specifies the CA Panvalet library DASD type.
	These are the valid values:
	`FBA`
	`3331` (3330-11 disk devices)
	`3340`
	`3350`
	`3375`
	`3380` Default
	`3390`

PanDtf	(CA Panvalet library file name) Category: JCL Type: Local Reload requirement: `ZRELOAD GENOPTS` Required. DD or name of the first or only CA Panvalet library Zeke uses to retrieve JCL (up to seven alphanumeric characters long). A DD or statement with the same name must reference the Panvalet library Zeke accesses. The default value is `OMITTED`.

Option	Description
PanSys	(CA Panvalet library logical unit) VSE only Category: JCL Type: Local Reload requirement: `ZRELOAD GENOPTS` Required. Logical unit used to access the CA Panvalet library (up to six alphanumeric characters long). This is the logical unit in the extent statement that follows the DLBL for the Panvalet library. This logical unit must be assigned, in the VSE/POWER partition, to the device where the Panvalet library resides.
PauseEoj	(Pause cancelled partition) VSE only Category: Dispatching Type: Local Reload requirement: `ZKILL TRACK` Required. Indicates whether Zeke pauses a partition that cancels during program execution.
PbTrack	(Problem tracking) Category: Exits Type: Local Reload requirement: `ZKILL TRACK` Required. Indicates whether you want to use the problem tracking facility. These are the valid values: `Y` Using problem tracking facility. `N` Default. Not using problem tracking facility.
PctWarn	(Database usage percentage warning) Category: Messages Type: Global Reload requirement: `ZRELOAD GENOPTS` Required. Indicates the percentage of database usage that must reached before a warning message is issued to the console. The valid values range from `01` through `99`. The default value is `80` (i.e., when the Zeke database is 80 percent full, Zeke issues a message to the console until the percent full goes below 80 percent).

Option	Description
PdsDD	(PDS ddname) z/OS only Category: JCL Type: Global Reload requirement: `ZRELOAD GENOPTS` Required. Default ddname (up to eight alphanumeric characters long) of the partitioned dataset to use (if no name is specified on the EMR) for a new job that uses a PDS for the JCL repository. The default ddname is `ZEKEJCL`.
PendInv	Category: Dispatching Type: Global Reload requirement: `ZRELOAD GENOPTS` Required. Time (in minutes) Zeke waits before dispatching a pending event to an initiator again. At the end of this time interval, Zeke adds the initiator back to the table of available /initiators. These are the valid values: `0` — No interval. The event will remain pending until an initiator is available and the event can be started (or until an operator intervenes). *nnn* — Specifies the time interval (in minutes). The valid values range from `001` through `999`.
PendMsg	(Pending event messages) Category: Messages Type: Local Reload requirement: `ZRELOAD GENOPTS` Required. Specifies the interval in minutes (*mmm*) between messages that notify of a pending event. Specify `000` if you do not want messages to be issued.
PJclPrt	(Print POWER JCL statements) VSE only Category: Messages Type: Local Reload requirement: `ZRELOAD GENOPTS` Required. Indicates whether to display POWER JCL statements at the console, and also print them to the job log for debugging purposes.

Option	Description

Posid	(Positively identify Zeke jobs)
	Category: JCL
	Type: Global
	Reload requirement: `ZKILL TRACK`
	Indicates whether to assign a unique POSID (positive event ID) to each Zeke-controlled job event. The job is tracked by its jobname and assigned POSID until the job is done.
	Note:
	All Zeke systems that share the same database must have the same Posid value.
	These are the valid values:
	`N` — Do not assign a unique ID and track jobs by jobname only.
	Note:
	Because a POSID cannot be used for an event with the JCL sources CMS or JCLMAN, you must set POSID to `N` or set the Control field in the EMR to `N` when submitting from these sources.
	`Y` — Default. Assign a unique POSID (positive event ID) to each Zeke job event. The job is tracked by its jobname and assigned POSID until the job is done.
	If this option is set to `Y`, Zeke inserts an IEFBR14 step with the POSID information either after the job card or as the last step in each Zeke-controlled job event. For example:
	`//* THE ZEKECTL STEP IS INSERTED BY ZEKE.`
	`//ZEKECTL  EXEC`
	`  PGM=IEFBR14,COND=ONLY,PARM=(A913EC42,199915C,00000012`
	`//A9BD957F,'MEDADVLP',LDVLP,'DVLPZEKE')`
	The placement of the POSID information is determined by the Posidend generation option.

Option	Description
	Note: For multiple event versions, the version number is passed to the submitting Zeke system as part of the POSID information. If the submitting system also supports multiple event versions, the version number enables the dispatching system to correctly identify the associated SQR. **Note:** Zeke does not support multiple jobs per event. When this option is set to `Y`, only the first job of an event is considered a Zeke job. All other jobs in the same event are considered nonZeke jobs; therefore, each job should be defined in a separate event. **Note:** To override the Posid option for a specific event, use the Control field on the EMR. When Control is set to `N`, Zeke does not recognize the event as Zeke-controlled, and marks the job as EOJ at dispatch. If you update this option, you must re-cycle Zeke.
PosidEnd	(Posid information placement) Category: JCL Type: Global Reload requirement: `ZRELOAD GENOPTS` Required. Indicates whether POSID information is placed at the end or the beginning of Zeke-controlled jobs. This option is valid only if Posid (page 566) is set to `Y`. **Note:** All Zeke systems that share the same database must have the same Posidend value. These are the valid values: `N` — Default. POSID information is placed at the start of the job (as the first step).

Option	Description	
	Y	POSID information is placed at the end of the job (as the last step). When PosidEnd is set to Y, Zeke jobs that fail with JCL errors will not reach a FAIL status, but will remain pending.
	For most jobs, the POSID is acceptable in either location; however, there are cases where one location could be preferable over the other. For example, if a job has the INCLUDE statement immediately following the job card and the included member has any JCL statements that must precede the first step (e.g, JOBLIB), then the job will fail if Posidend is set to N. If a job has in-stream data (i.e., SYSIN DD *) containing an imbedded job, then the job will fail if Posidend is set to Y. If a remote job is dispatched to another Zeke system through OASIS/DMS, then the Posidend generation option on the execution system governs POSID placement. If a remote job is dispatched to another system through NJE, then the dispatching Zeke's Posidend generation option governs POSID placement.	
PriLate	(Late event priority) Category: Dispatching Type: Global Reload requirement: ZRELOAD GENOPTS Required. Indicates whether to dispatch late events with a higher priority. These are the valid values:	
	Y	Dispatch late events before other events regardless of schedule time.
	N	Default. Dispatch events in schedule time sequence (without regard for the late time).

Option	Description
RdrExit	(POWER reader exit) VSE only Category: Exits Type: Local Reload requirement: `ZKILL COLD` Optional. Name of your VSE/POWER reader exit to Zeke (up to eight alphanumeric characters long). Zeke will CDLOAD this phase at system startup and branch to it when Zeke's reader exit has finished processing. **Note:** Do not use this parameter if Zeke is running under Software Pursuit's SPRI spooling system.
RefEvent	(Event references) Category: Scheduling Type: Global Reload requirement: `ZRELOAD GENOPTS` Indicates whether REFEVENT references from one EMR to another EMR by *event number* are allowed. A REFEVENT reference defined in an EMR schedules the event according to the calendar (including nonworking days) and OCCURS clause of another event. You can specify the event that you want to reference either by event number or event name. **Caution!** Event numbers are unique; however, because Zeke assigns the event numbers automatically and can re-assign available, previously-used numbers to new events, ASG recommends you reference other events by *event name* to avoid unintended references. These are the valid values: `A` (Allow) Default. REFEVENT references from one event to another by event number are valid. `F` (Fail) A REFEVENT reference from one event to another by event number is not allowed. The event is not scheduled. `W` (Warn) Issue a warning when a REFEVENT reference from one event to another by event number is found. The event is scheduled normally.

Option	Description
RemovDQ	(Remove from dispatch queue) Category: Dispatching Type: Global Reload requirement: `ZRELOAD GENOPTS` Required. Indicates whether events can be removed from the dispatch queue if they are no longer TIME and/or WHEN satisfied. These are the valid values: `Y` — Default. Events can be removed from the dispatch queue if they are no longer TIME and/or WHEN satisfied. `N` — Events cannot be removed from the dispatch queue if they are no longer TIME and/or WHEN satisfied. **Note:** If a ZALTER WHENOK command is issued against an event, it will remain in the dispatch queue, regardless of the RemovDQ setting.
RemTrig	(Remote triggering) Category: Dispatching Type: Global Reload requirement: `ZRELOAD GENOPTS` 2-character code that indicates how to handle a remote trigger for multiple SQRs in the schedule (i.e., if the remote trigger does not contain a schedule or run date). (If the remote trigger has a schedule and run date, this option is ignored and the Trigdt option is applied to the dates to process the trigger.) These are the valid values: `ND` — Trigger the event with the newest schedule/run date only. `NT` — If multiple dates exist, then *do not* trigger any event. If only one date exists, trigger that event. `OD` — Trigger the event with the oldest schedule/run date only. `TA` — Trigger all events.

Option	Description
	Note: If the remote trigger was satisfied by a Zeke-controlled job, the SQR's schedule and run dates are sent with the trigger. If the remote trigger was satisfied by a non-Zeke job on a Zeke system, the system's current date is sent as the schedule date and run date with the trigger.
RepCls	(Replace class) VSE only Category: JCL Type: Global Reload requirement: `ZRELOAD GENOPTS` Required. Indicates whether to scan each submitted jobstream for the class on VSE/POWER job statements. Zeke changes the class to match the VSE/POWER class being used to submit the job. Affects only locally submitted jobs.
RepJGrp	(Replace GROUP=) z/OS only Category: JCL Type: Global Reload requirement: `ZRELOAD GENOPTS` **Note:** RACF surrogate processing must be set up before this option can be used. See the *ASG-Zeke Scheduling for z/OS Installation Guide* for details. Required. Indicates whether to replace the GROUP= keyword on the JOB card with the Secgroup value from the event's EMR. **Note:** RepJGrp processing is done just *before* the job is submitted, after any user exits have been invoked. These are the valid values: `N` (Never) Default. Do not modify the GROUP= keyword on the JOB card. `A` (Always) If the JOB card already has a GROUP= value, replace it with the Secgroup value from the EMR; if the JOB card does not have a GROUP= keyword, add it with the Secgroup value from the EMR. If a Secgroup value is not specified in the EMR, the JOB card is not modified.

Option	Description	
	C	(Conditional) If the JOB card already has a GROUP= value, do not modify it; if the JOB card does not have a GROUP= keyword, add it with the Secgroup value from the EMR. If a Secgroup value is not specified in the EMR, the JOB card is not modified.
RepJName	(Replace jobname) Category: JCL Type: Global Reload requirement: `ZRELOAD GENOPTS` Required. Indicates whether to replace the job name on the JOB card with the jobname from the event's EMR. **Note:** RepJName processing is done *before* any user exits have been invoked. These are the valid values:	
	N	Default. Do not modify the jobname on the JOB card.
	Y	Replace the jobname on the JOB card. (Only the first JOB card in the JCL is modified.) If the jobname from the EMR is longer than the jobname on the JOB card, Zeke makes room for the longer name by splitting the card at a comma that separates operands.

Option	Description
RepJSEnv	(Replace scheduling environment) z/OS only Category: JCL Type: Global Reload requirement: `ZRELOAD GENOPTS` This option controls whether to allow Zeke to insert (or replace) the SCHENV keyword on the job card before the job event is submitted to JES. (Zeke retrieves the SCHENV value from the SQR.) Depending on how you set this value, the SCHENV value could differ in a job event's SQR and its JCL. In this case, Zeke schedules and dispatches the event according to the environment specified in the SQR. If a different environment is specified on the job card, and is undefined or inactive, the job is held in the JES input queue or fails with a JCL error. If a job event has a JCL source of JESQ, Zeke does not modify the job card. then a RepJSEnv value of `N` is assumed. **Note:** RepJSEnv processing is done just before the job is submitted, after any user exits have been invoked. These are the valid values: `N` (Never) Default. Zeke does not insert/replace the SCHENV keyword on the job card. `A` (Always) Zeke always inserts/replaces the SCHENV keyword on the job card. `C` (Conditional) Zeke inserts the SCHENV keyword on the job card only if no keyword exists. Zeke does not replace an existing keyword.
RepJUser	(Replace user ID) z/OS only Category: JCL Type: Global Reload requirement: `ZRELOAD GENOPTS` Required. Indicates whether to replace the USER= keyword on the JOB card with the user ID from the event's EMR. RACF surrogate processing must be set up before this option can be used. See the *ASG-Zeke Scheduling for z/OS Installation Guide* for details. These are the valid values:

Option	Description	
	N	(Never) Default. Do not modify the USER= keyword on the JOB card.
	A	(Always) If the JOB card already has a USER= value, replace it with the user ID value from the EMR; if the JOB card does not have a USER= keyword, add it with the user ID value from the EMR. If a user ID is not specified in the EMR, the JOB card is not modified.
	C	(Conditional) If the JOB card already has a USER= value, do not modify it; if the JOB card does not have a USER= keyword, add it with the user ID value from the EMR. If a user ID is not specified in the EMR, the JOB card is not modified.
	Note: RepJUser processing is done just before the job is submitted, after any user exits have been invoked.	
ReqOpid	(Require operator ID) Category: Security Type: Global Reload requirement: `ZRELOAD GENOPTS` Required. Indicates whether an authorized Operator ID is required to log on to Zeke. These are the valid values:	
	Y	Unauthorized users cannot log on. **Caution!** Before changing this option to Y, you must have an authorized operator ID with WRITE access to security. Otherwise, all users can be locked out of the online facility.
	N	Default. Unauthorized users can log on and are assigned the default operator ID entered in the Defopid field.

Option	Description
Retain	(Retain events) Category: Scheduling Type: Global Reload requirement: `ZRELOAD GENOPTS` Required. Indicates whether to retain events that have not been dispatched. Zeke uses this value as the default when adding an EMR. These are the valid values: `Y` Default. Retain non-dispatched events during the next schedule run. `N` Discard non-dispatched events during the next schedule run.
RetDays	(Days to retain events) Category: Scheduling Type: Global Reload requirement: `ZRELOAD GENOPTS` (Required only if Retpend is set to `Y`.) Specifies the number of days (*ddd*) to retain pending or abend job events. Zeke discards the schedule records after this number of days. The default value is `002`. For example, a value of `000` discards the events the next schedule run; a value of `001` discards the events after two schedule runs.
RetDone	(Retain completed events) Category: Scheduling Type: Global Reload requirement: `ZRELOAD GENOPTS` Required. Indicates whether to retain completed events in the schedule. These are the valid values: `Y` Default. Retain events even after EOJ. `N` Discard events after EOJ.

Option	Description
RetPend	(Retain pending events) Category: Scheduling Type: Global Reload requirement: `ZRELOAD GENOPTS` Required. Indicates whether to retain uncompleted Schedule Queue Records across SCHEDULE function runs. This is for events that have been dispatched, yet have not reached a DONE status when it is time to run the SCHEDULE function for the next day. These are the valid values: `Y` Default. Retain schedule records until a normal EOJ is recorded or the job is disabled. `N` Delete dispatched schedule records when the SCHEDULE function runs with a new date.
ScomMax	(SCOM maximum subtasks) VSE only Category: Dispatching Type: Local Reload requirement: `ZKILL TRACK` Required. Maximum number of subtasks allowed for a system command event (SCOM). A subtask is required for each scheduled SCOM event that is processing concurrently. If an SCOM event is dispatched and there is no available subtask, the event is delayed until a POWER subtask is available. The valid values range from `01` through `99`. The default value is `01`.
ScomWt	(SCOM maximum wait) VSE only Category: Dispatching Type: Local Reload requirement: `ZRELOAD GENOPTS` Required. Maximum number of minutes that an SCOM can wait to be issued before timing out. The valid values range from `01` through `99`. The default value is `05`.

Option	Description
SecExitw	(Security exit storage) Category: Exits Type: Global Reload requirement: `ZRELOAD GENOPTS` Optional. 4-character numeric value that specifies the number of bytes of storage allocated and passed to the Zeke security exit routine (ZEKE15B).
SecFail	(Security verification failure) Category: Security Type: Global Reload requirement: `ZRELOAD GENOPTS` Required. Indicates the action for Zeke to take if batch security verification fails. These are the valid values: `C` Default. Cancel the job. `I` Ignore the request, but continue processing.
SecHide1	(Security for primary database records) Category: Security Type: Global Reload requirement: `ZRELOAD GENOPTS` Required. Indicates whether to hide primary database records if security access is denied. These are the valid values: `Y` Display only those records the user has access to. Any Zeke online screen that displays a directory of records will only display those records to which the user has at least Read access. `N` Default. Display all security records, regardless of security access.

Option	Description
SecHide2	(Security for subordinate database records) Category: Security Type: Global Reload requirement: `ZRELOAD GENOPTS` Required. Indicates whether to hide subordinate database records if security access is denied. These are the valid values: `Y` — Display only those records the user has access to. Any Zeke online screen that displays an asterisk (*) to indicate a sub-record exists will only display those records to which the user has at least Read access. `N` — Default. Display all security records, regardless of security access.
SecSel	(Security checks using selection criteria) Category: Security Type: Global Reload requirement: `ZRELOAD GENOPTS` Required. Indicates whether to perform security verification on a group of records using selection criteria or to check each individual record. These are the valid values: `Y` — Default. Perform security check using selection criteria. `N` — Perform security verification for individual records.
SecUInit	(Security field auto-fill) z/OS only Category: Security Type: Global Reload requirement: `ZRELOAD GENOPTS` Indicates whether the Usrid and Secgroup fields on the EMR are automatically populated (by SAF) with the user ID and security group of the user that created the EMR (including those created from OpsCentral).

Option	Description
	Note: For EMRs added via the batch interface, the Usrid and Secgroup fields are *not* populated by SAF. These are the valid values: `N` Default. SAF does not populate the user ID and security group fields. `Y` SAF populates the EMR with the user ID and security group of the user who created the EMR. (This overrides any values entered by the user or obtained from the event template for these fields.)
SecULock	(Security field update) Category: Security Type: Global Reload requirement: `ZRELOAD GENOPTS` Indicates whether online users (including OpsCentral users) are permitted to update the Usrid and Secgroup fields on the EMR. **Note:** These fields can *always* be modified using the batch interface, regardless of the SecULock setting. These are the valid values: `E` (ESI-controlled) This option uses the OASIS External Security Interface (ESI), and is therefore valid only for z/OS systems that have ESI Actv set to `Y`. If ESI denies Write access (based on the Z$ACCESS class), the fields are read-only; otherwise, they can be updated. If ESI is not active (i.e., if ESI Actv is set to `N`), this option is treated as if SecULock is set to `N`. `N` Default. Do not lock these fields as read-only fields. `Y` Lock these fields as read-only fields.

Option	Description
Steplog	(Log step names) VSE only Category: Messages Type: Local Reload requirement: `ZKILL` Required. Indicates whether to log Zeke step names on SYSLOG.
StepLst	(List step names) VSE only Category: Messages Type: Local Reload requirement: `ZKILL` Required. Indicates whether to log Zeke step names on SYSLST.
SubData	(Variable substitution) Category: Variables Type: Global Reload requirement: `ZRELOAD GENOPTS` Required. Indicates whether to perform variable substitution for JCL and non-JCL statements. These are the valid values: `Y` Substitute variables at submittal. Zeke scans all statements as the jobstream is submitted through JES. If a variable is found, it is replaced with its current value. `N` Default. Do not substitute variables.

Option	Description
SvMinBrk	(Schedule View Minimum Breakout) z/OS only Category: User Interface Type: Local Reload requirement: `ZRELOAD GENOPTS` Required. Indicates the minimum interval at which Schedule View issues a terminal read to determine whether to break out of auto-update mode. Use this option to limit the frequency that the AUTO command updates displayed data and checks for a 'breakout' request. **Notes:** • Enforcing a higher limit reduces the number of terminal reads by Schedule View operators site-wide, but does also increase the time before an operator breakout request is acknowledged by the auto-update process. • Changes to this option do not affect the current operator until the next time the operator issues the INT or AUTO command. These are the valid values: `0` — No minimum. *nn* — Minimum value (ranging from `1` through `99`) allowed to be specified for the breakout (*`wait`*) operand on the INT command. The actual minimum may vary depending on the current operator's breakout interval.
SvMinRef	(Schedule View Minimum Data Refresh) z/OS only Category: User Interface Type: Local Reload requirement: `ZRELOAD GENOPTS` Required. Indicates the minimum interval at which Schedule View scans the list of schedule events for updates while in auto-update mode. **Notes:** • Updating event statuses could use a noticeable amount of system resources depending on the number of events, number of Schedule View operators, and the operators' display filters. Limiting the frequency at which Schedule View is updated reduces system resource consumption, but also increases the time that displayed schedule data may be outdated. • Changes to this option do not affect a current operator until the next time the operator issues the INT or AUTO command.

Option	Description
	These are the valid values: `0` No minimum. *nn* Minimum value (ranging from `1` through `99`) allowed for the 'data refresh' (*refresh*) operand on the INT command.
TapeIO	(Tape I/Os) Category: Dispatching Type: Local Reload requirement: `ZKILL TRACK` Required. Specifies the number (*nnn*) of start I/Os that are required before Zeke recognizes a tape drive as being accessed. The default value is `100`.
TempJob	(Default job event template) Category: User interface Type: Global Reload requirement: `ZRELOAD GENOPTS` Name of the default template to use as a model when creating job events from the Add Event Master Record Function screen (up to 12 alphanumeric characters long). The name entered here appears in the Template field on the Add Event Master Record Function screen.
TempMsg	(Default message event template) Category: User interface Type: Global Reload requirement: `ZRELOAD GENOPTS` Name of the default template to use as a model when creating message events from the Add Event Master Record Function screen (up to 12 alphanumeric characters long). The name entered here appears in the Template field on the Add Event Master Record Function screen.

Option	Description
TempPcom	(Default POWER command event template) VSE only Category: User interface Type: Global Reload requirement: `ZRELOAD GENOPTS` Name of the default template to use as a model when creating PCOM events from the Add Event Master Record Function screen (up to 12 alphanumeric characters long). The name entered here appears in the Template field on the Add Event Master Record Function screen.
TempRexx	(Default REXX event template) z/OS only Category: User interface Type: Global Reload requirement: `ZRELOAD GENOPTS` Name of the default template to use as a model when creating REXX events from the Add Event Master Record Function screen (up to 12 alphanumeric characters long). The name entered here appears in the Template field on the Add Event Master Record Function screen.
TempScom	(Default SCOM command event template) Category: User interface Type: Global Reload requirement: `ZRELOAD GENOPTS` Name of the default template to use as a model when creating SCOM events from the Add Event Master Record Function screen (up to 12 alphanumeric characters long). The name entered here appears in the Template field on the Add Event Master Record Function screen.
TempTask	(Default TASK event template) Category: User interface Type: Global Reload requirement: `ZRELOAD GENOPTS` Name of the default template to use as a model when creating TASK events from the Add Event Master Record Function screen (up to 12 alphanumeric characters long). The name entered here appears in the Template field on the Add Event Master Record Function screen.

Option	Description
TempVcom	(Default VCOM command event template) Category: User interface Type: Global Reload requirement: `ZRELOAD GENOPTS` Name of the default template to use as a model when creating VCOM events from the Add Event Master Record Function screen (up to 12 alphanumeric characters long). The name entered here appears in the Template field on the Add Event Master Record Function screen.
TempWork	(Default work center event template) Category: User interface Type: Global Reload requirement: `ZRELOAD GENOPTS` Name of the default template to use as a model when creating WORK events from the Add Event Master Record Function screen (up to 12 alphanumeric characters long). The name entered here appears in the Template field on the Add Event Master Record Function screen.
TempZcom	(Default ZCOM command event template) Category: User interface Type: Global Reload requirement: `ZRELOAD GENOPTS` Name of the default template to use as a model when creating ZCOM events from the Add Event Master Record Function screen (up to 12 alphanumeric characters long). The name entered here appears in the Template field on the Add Event Master Record Function screen.
Tr01Aenq	(Trace messaging) z/OS only Category: Traces Type: Local Reload requirement: `ZRELOAD GENOPTS` Indicates whether ZEKE01A enqueue/dequeue trace messages are written to the data space log. Activating this trace option increases CPU usage and could slow Zeke processing as a result. It also writes a large number of trace message to the data space log. If you have data space off-loading set up to take place automatically, keep in mind that the increase in messages being generated will cause the log datasets to fill more quickly.

Option	Description
	These are the valid values: `N` Do not write ZEKE01A enqueue/dequeue trace messages to the data space log. `Y` Default. Write ZEKE01A enqueue/dequeue trace messages to the data space log.
Trfcomtr	(Trace messaging) z/OS only Category: Traces Type: Local Reload requirement: `ZRELOAD GENOPTS` Indicates whether database communications trace messages are written to the data space log. These are the valid values: `N` Do not write database communications trace messages to the data space log. `Y` Default. Write database communications trace messages to the data space log.
Trfdnall	(Trace messaging) z/OS only Category: Traces Type: Local Reload requirement: `ZRELOAD GENOPTS` Indicates whether all dataname trace messages are written to the data space log. These are the valid values: `N` Do not write dataname trace messages to the data space log. `Y` Default. Write all dataname trace messages to the data space log.

Option	Description
Trfdnerr	(Trace messaging) z/OS only Category: Traces Type: Local Reload requirement: `ZRELOAD GENOPTS` Indicates whether dataname error trace messages are written to the data space log. These are the valid values: `N` Do not write dataname error trace messages to the data space log. `Y` Default. Write dataname error trace messages to the data space log.
Trfjcltr	(Trace messaging) z/OS only Category: Traces Type: Local Reload requirement: `ZRELOAD GENOPTS` Indicates whether all JCL trace messages are written to the data space log. These are the valid values: `N` Default. Do not write JCL trace messages to the data space log. `Y` Write all JCL trace messages to the data space log.
Trfjclwt	(Trace messaging) z/OS only Category: Traces Type: Local Reload requirement: `ZRELOAD GENOPTS` Indicates whether JCL write trace messages are written to the data space log. These are the valid values: `N` Default. Do not write JCL write trace messages to the data space log. `Y` Write JCL write trace messages to the data space log.

Option	Description
Trfsecex	(Trace messaging) z/OS only Category: Traces Type: Local Reload requirement: `ZRELOAD GENOPTS` Indicates whether security expand trace messages are written to the data space log. These are the valid values `N` Default. Do not write security expand trace messages to the data space log. `Y` Write security expand trace messages to the data space log.
Trfsectr	(Trace messaging) z/OS only Category: Traces Type: Local Reload requirement: `ZRELOAD GENOPTS` Indicates whether all security trace messages are written to the data space log. These are the valid values `N` Do not write security trace messages to the data space log. `Y` Default. Write all security trace messages to the data space log.
Trfsmfm	(Trace messaging) z/OS only Category: Traces Type: Local Reload requirement: `ZRELOAD GENOPTS` Indicates whether SMF exit trace messages are written to the data space log. These are the valid values `N` Default. Do not write SMF exit trace messages to the data space log. `Y` Write SMF exit trace messages to the data space log.

Option	Description
Trfucbsc	(Trace messaging) z/OS only Category: Traces Type: Local Reload requirement: `ZRELOAD GENOPTS` Indicates whether UCB scan trace messages are written to the data space log. These are the valid values `N` Default. Do not write UCB scan trace messages to the data space log. `Y` Write UCB scan trace messages to the data space log.
TrigDt	(Date triggering) Category: Dispatching Type: Global Reload requirement: `ZRELOAD GENOPTS` Required. Single-character code that indicates whether a Zeke-controlled job can trigger an event's WHEN condition if the dates are different; and, if not, whether to use the schedule date or the run date. These are the valid values: `A` Default. Any job (by the name) can trigger an event's WHEN condition regardless of the schedule dates. `R` *Run dates* must be the same for a job to trigger another event's WHEN condition. The run date is the date that the event was added to the schedule (i.e., by the ZADD command or the SCHEDULE batch function). `S` *Schedule dates* must be the same for a job to trigger another event's WHEN condition. The schedule date is the date that an event would have run if it were not a nonworking day (weekend or holiday).

Option	Description
	Note: If multiple Zeke systems are sharing a database (i.e., the MultSys generation option is set to Y), each of the Zeke systems must be set to the same Trigdt value. Otherwise, you might experience excessive database I/O.
TrigJob	(Job triggering) Category: Dispatching Type: Global Reload requirement: ZRELOAD GENOPTS Required. Single-character code that indicates whether a job can satisfy another event's WHEN conditions if the job is not dispatched by this Zeke (or by another Zeke sharing the same database). These are the valid values: A — Default. Any job can satisfy triggers on this Zeke (regardless of how/where the job is dispatched). C — Only jobs dispatched by this Zeke (or by a Zeke sharing the same database) can satisfy triggers on this Zeke. Remote Zeke jobs and non-Zeke jobs are ignored. S — Only jobs dispatched by this Zeke (or by a Zeke sharing the same database) and non-Zeke jobs can satisfy triggers on this Zeke. Remote Zeke jobs are ignored. **Note:** When a remote trigger is satisfied by a job on a remote Zeke system or Zeke Services agent (that has its Netregid specified as the object of the AT keyword defined in the remote trigger), a non-Zeke job as well as Zeke-controlled job will satisfy the trigger, regardless of the setting of the Zeke generation option TrigJob. Both the originating and remote Zeke systems ignore the TrigJob generation option when satisfying remote triggers.

Option	Description
TrigRrn	(Rerun job triggering) Category: Scheduling Type: Global Reload requirement: `ZRELOAD GENOPTS` Required. Indicates whether a job with a rerun designation can trigger an event's WHEN condition. The rerun designation is specified for a schedule record by the ZADD, ZALTER, and ZREFRESH commands. These are the valid values: `Y` Any job can trigger an event's WHEN condition regardless of the rerun designation. `N` Default. A rerun job cannot trigger another event's WHEN condition.
Trplxf1a	(Trace messaging) z/OS only Category: Traces Type: Local Reload requirement: `ZRELOAD GENOPTS` Indicates whether XCF work check trace messages are written to the data space log. These are the valid values: `N` Default. Do not write XCF work check trace messages to the data space log. `Y` Write XCF work check trace messages to the data space log.
Trplxf1b	(Trace messaging) z/OS only Category: Traces Type: Local Reload requirement: `ZRELOAD GENOPTS` Indicates whether XCF work item trace messages are written to the data space log. These are the valid values: `N` Default. Do not write XCF work item trace messages to the data space log.

Option	Description	
	Y	Write XCF work item trace messages to the data space log.
Trplxf1c	(Trace messaging) z/OS only Category: Traces Type: Local Reload requirement: ZRELOAD GENOPTS Indicates whether XCF broadcast trace messages are written to the data space log. These are the valid values:	
	N	Default. Do not write XCF broadcast trace messages to the data space log.
	Y	Write XCF broadcast trace messages to the data space log.
Trplxf1d	(Trace messaging) z/OS only Category: Traces Type: Local Reload requirement: ZRELOAD GENOPTS Indicates whether XCF Zeke up/down trace messages are written to the data space log. These are the valid values:	
	N	Default. Do not write XCF Zeke up/down trace messages to the data space log.
	Y	Write XCF Zeke up/down trace messages to the data space log.
Trplxf1e	(Trace messaging) z/OS only Category: Traces Type: Local Reload requirement: ZRELOAD GENOPTS Indicates whether XCF system up/down trace messages are written to the data space log. These are the valid values:	

Option	Description	
	N	Default. Do not write XCF system up/down trace messages to the data space log.
	Y	Write XCF system up/down trace messages to the data space log.
Trplxf1f	(Trace messaging) z/OS only Category: Traces Type: Local Reload requirement: `ZRELOAD GENOPTS` Indicates whether XCF cleanup trace messages are written to the data space log. These are the valid values:	
	N	Default. Do not write XCF cleanup trace messages to the data space log.
	Y	Write XCF cleanup trace messages to the data space log.
Trplxf1g	(Trace messaging) z/OS only Category: Traces Type: Local Reload requirement: `ZRELOAD GENOPTS` Indicates whether XCF initialization trace messages are written to the data space log. These are the valid values:	
	N	Default. Do not write XCF initialization trace messages to the data space log.
	Y	Write XCF initialization trace messages to the data space log.
Trplxf1h	(Trace messaging) z/OS only Category: Traces Type: Local Reload requirement: `ZRELOAD GENOPTS` Indicates whether XCF polling trace messages are written to the data space log.	

Option	Description
	These are the valid values: `N` Default. Do not write XCF polling trace messages to the data space log. `Y` Write XCF polling trace messages to the data space log.
Trplxf2i	(Trace messaging) z/OS only Category: Traces Type: Local Reload requirement: `ZRELOAD GENOPTS` Indicates whether XCF communication record trace messages are written to the data space log. These are the valid values: `N` Default. Do not write XCF communication record trace messages to the data space log. `Y` Write XCF communication record trace messages to the data space log.
Trplxf2j	(Trace messaging) z/OS only Category: Traces Type: Local Reload requirement: `ZRELOAD GENOPTS` Indicates whether XCF communication record work trace messages are written to the data space log. These are the valid values: `N` Default. Do not write XCF communication record work trace messages to the data space log. `Y` Write XCF communication record work trace messages to the data space log.

Option	Description
Trplxf2k	(Trace messaging) z/OS only Category: Traces Type: Local Reload requirement: `ZRELOAD GENOPTS` Indicates whether XCF work item trace messages are written to the data space log. These are the valid values: `N` Default. Do not write XCF work item trace messages to the data space log. `Y` Write XCF work item trace messages to the data space log.
Trplxf2l	(Trace messaging) z/OS only Category: Traces Type: Local Reload requirement: `ZRELOAD GENOPTS` Indicates whether XCF cleanup messages are written to the data space log. These are the valid values: `N` Default. Do not write XCF cleanup messages to the data space log. `Y` Write XCF cleanup messages to the data space log.
Trplxf2m	(Trace messaging) z/OS only Category: Traces Type: Local Reload requirement: `ZRELOAD GENOPTS` Indicates whether new communication record trace messages are written to the data space log. These are the valid values: `N` Default. Do not write new communication record trace messages to the data space log. `Y` Write new communication record trace messages to the data space log.

Option	Description
Trplxf2n	(Trace messaging) z/OS only Category: Traces Type: Local Reload requirement: `ZRELOAD GENOPTS` Indicates whether XCF broadcast trace messages are written to the data space log. These are the valid values: `N` Default. Do not write XCF broadcast trace messages to the data space log. `Y` Write XCF broadcast trace messages to the data space log.
Trplxf2o	(Trace messaging) z/OS only Category: Traces Type: Local Reload requirement: `ZRELOAD GENOPTS` Indicates whether XCF communication record trace messages are written to the data space log. These are the valid values: `N` Default. Do not write XCF communication record trace messages to the data space log. `Y` Write new XCF communication record trace messages to the data space log.
Trplxf2p	(Trace messaging) z/OS only Category: Traces Type: Local Reload requirement: `ZRELOAD GENOPTS` Indicates whether XCF schedule record trace messages are written to the data space log. These are the valid values: `N` Default. Do not write XCF schedule record trace messages to the data space log.

Option	Description	
	Y	Write XCF schedule record trace messages to the data space log.
Trplxf3q	(Trace messaging) z/OS only	
	Category: Traces	
	Type: Local	
	Reload requirement: `ZRELOAD GENOPTS`	
	Indicates whether XCF schedule record trace messages are written to the data space log.	
	These are the valid values:	
	N	Default. Do not write XCF schedule record trace messages to the data space log.
	Y	Write XCF schedule record trace messages to the data space log.
Trplxf3r	(Trace messaging) z/OS only	
	Category: Traces	
	Type: Local	
	Reload requirement: `ZRELOAD GENOPTS`	
	Indicates whether XCF communication queue trace messages are written to the data space log.	
	These are the valid values:	
	N	Default. Do not write XCF communication queue trace messages to the data space log.
	Y	Write XCF communication queue trace messages to the data space log.
Trplxf3s	(Trace messaging) z/OS only	
	Category: Traces	
	Type: Local	
	Reload requirement: `ZRELOAD GENOPTS`	
	Indicates whether XCF get trace messages are written to the data space log.	
	These are the valid values:	

Option	Description	
	N	Default. Do not write XCF get trace messages to the data space log.
	Y	Write XCF get trace messages to the data space log.
Trplxf3t	(Trace messaging) z/OS only Category: Traces Type: Local Reload requirement: `ZRELOAD GENOPTS` Indicates whether XCF put trace records are written to the data space log. These are the valid values:	
	N	Do not write XCF put trace records to the data space log.
	Y	Default. Write XCF put trace records to the data space log.
Trplxf3u	(Trace messaging) z/OS only Category: Traces Type: Local Reload requirement: `ZRELOAD GENOPTS` Indicates whether XCF weak satisfy trace messages are written to the data space log. These are the valid values:	
	N	Do not write XCF weak satisfy trace messages to the data space log.
	Y	Default. Write XCF weak satisfy trace messages to the data space log.
Trplxf3v	(Trace messaging) z/OS only Category: Traces Type: Local Reload requirement: `ZRELOAD GENOPTS` Indicates whether ETT extra read trace messages are written to the data space log. These are the valid values:	

Option	Description	
	N	Default. Do not write ETT extra read trace messages to the data space log.
	Y	Write ETT extra read trace messages to the data space log.
Trplxf3w	(Trace messaging) z/OS only	
	Category: Traces	
	Type: Local	
	Reload requirement: ZRELOAD GENOPTS	
	Indicates whether satisfy trigger trace messages are written to the data space log.	
	These are the valid values:	
	N	Do not write satisfy trigger trace messages to the data space log.
	Y	Default. Write satisfy trigger trace messages to the data space log.
Trplxf3x	(Trace messaging) z/OS only	
	Category: Traces	
	Type: Local	
	Reload requirement: ZRELOAD GENOPTS	
	Indicates whether cross-system JCL error trace messages are written to the data space log.	
	These are the valid values:	
	N	Default. Do not write cross-system JCL error trace messages to the data space log.
	Y	Write cross-system JCL error trace messages to the data space log.
Trplxf4y	(Trace messaging) z/OS only	
	Category: Traces	
	Type: Local	
	Reload requirement: ZRELOAD GENOPTS	
	Indicates whether XCF resource trace messages are written to the data space log.	

Option	Description
	These are the valid values: `N` Default. Do not write XCF resource trace messages to the data space log. `Y` Write XCF resource to the data space log.
Trplxf4z	(Trace messaging) z/OS only Category: Traces Type: Local Reload requirement: `ZRELOAD GENOPTS` Indicates whether XCF resource trace messages are written to the data space log. These are the valid values: `N` Default. Do not write XCF resource trace messages to the data space log. `Y` Write XCF resource to the data space log.
Trt2dsl	(Trace messaging) z/OS only Category: Traces Type: Local Reload requirement: `ZRELOAD GENOPTS` Indicates whether schedule download trace messages are written to the data space log. These are the valid values: `N` Do not write schedule download trace messages to the data space log. `Y` Default. Write schedule download trace messages to the data space log.
Trt2jijo	(Trace messaging) z/OS only Category: Traces Type: Local Reload requirement: `ZRELOAD GENOPTS` Indicates whether JIJO trace messages are written to the data space log.

Option	Description
	These are the valid values: `N` Default. Do not write JIJO trace messages to the data space log. `Y` Write JIJO trace messages to the data space log.
Trt2notd	(Trace messaging) z/OS only Category: Traces Type: Local Reload requirement: `ZRELOAD GENOPTS` Indicates whether NOTDURING trace messages are written to the data space log. These are the valid values: `N` Do not write NOTDURING trace messages to the data space log. `Y` Default. Write NOTDURING trace messages to the data space log.
Trz0402i	(Trace messaging) z/OS only Category: Traces Type: Local Reload requirement: `ZRELOAD GENOPTS` Indicates whether Z0402I messages are written to the data space log. These are the valid values: `N` Do not write Z0402I messages to the data space log. `Y` Default. Write Z0402I messages to the data space log.
UsePJob	(Use POWER job cards) VSE only Category: JCL Type: Global Reload requirement: `ZRELOAD GENOPTS` Required. Indicates whether to use existing VSE/POWER job cards for the first job in the stream.

Option	Description
UserCls	(User class) Category: JCL Type: Global Reload requirement: `ZRELOAD GENOPTS` Required. Indicates how to determine the job submit class. These are the valid values: `Y` Use the class specified in the EMR to submit jobs to the system. `N` Default. Do not use the EMR class; enable Zeke to select initiators.
Userid	(User ID) Category: JCL Type: Global Reload requirement: `ZRELOAD GENOPTS` Name of the CMS machine that submits JCL (up to eight alphanumeric characters long). The default value is `ZEKE`.
U83supp	(SMF type 15 records) z/OS only Category: Exits Type: Local Reload requirement: `ZRELOAD GENOPTS` Required. Indicates whether Zeke require the operating system to suppress SMF type 15 records. When the record is suppressed, it is not written to the SMF dataset. These are the valid values: `Y` z/OS suppresses SMF type 15 records. `N` Default. z/OS does not suppress SMF type 15 records.

Option	Description
VmCmsin	(VM/CMS JCL wait) Category: JCL Type: Local Reload requirement: `ZRELOAD GENOPTS` (Required only if a non-VM system must obtain JCL from a CMS system.) Specifies the time in seconds (`ss`) that the ZEKE1 machine waits before checking for a CMS JCL request from a non-VM system. The default value is `10`.
VmJclid	(VM JCL user ID) Category: JCL Type: Local Reload requirement: `ZRELOAD GENOPTS` Required. Specifies the VM user ID to receive JCL from the ZEKE1 machine (up to eight alphanumeric characters long).
VmOperid	(VM message operator ID) Category: General Type: Local Reload requirement: `ZRELOAD GENOPTS` The VM operator ID (up to eight alphanumeric characters long) to receive message events. The default value is `OPERATOR`.
VollDtf	(CA Vollie library DTF name) VSE only Category: JCL Type: Local Reload requirement: `ZRELOAD GENOPTS` Required if using CA Vollie. The DTF name of the Vollie library Zeke uses to retrieve JCL (up to eight alphanumeric characters long). Add the DLBL statement to the Zeke start-up procedure. The default value is `OMITTED`.
VolLub	(CA Vollie library logical unit) VSE only Category: JCL Type: Local Reload requirement: `ZRELOAD GENOPTS` Required if using CA Vollie. The Vollie library logical unit. The valid values range from `000` through `254`. The default value is `000`.

Option	Description
VollXid	(CA Vollie library default OPIDENT) VSE only Category: JCL Type: Local Reload requirement: `ZRELOAD GENOPTS` Required if using CA Vollie. The default OPIDENT to use if the VOPIDENT is blank (up to three alphanumeric characters long). The default value is &&&.
WkTrgdn	(Weak event triggering—done) Category: Dispatching Type: Global Reload requirement: `ZKILL TRACK` Required. Indicates whether completed events (i.e., events in DONE status) can satisfy the WEOJ/WEOE (weak) WHEN conditions of other events in the schedule. These are the valid values: `Y` Previously completed (DONE) events can trigger events in the schedule that have weak WHEN conditions. `N` Default. Previously complete (DONE) events cannot trigger events in the schedule that have weak WHEN conditions.
WkTrgds	(Weak event triggering—disabled) Category: Dispatching Type: Global Reload requirement: `ZKILL TRACK` Required. Indicates whether disabled events can satisfy the WEOJ/WEOE (weak) WHEN conditions of other events in the schedule. These are the valid values:

Option	Description	
	Y	Disabled events can trigger events in the schedule that have weak WHEN conditions. **Note:** Disabling an event will result in the weak triggering of the dependent event. If the disabled event is enabled later, the weak trigger will no longer be satisfied. If multiple events satisfy the weak WHEN condition, *all* must be disabled for triggering to occur.
	N	Default. Disabled events cannot trigger events in the schedule that have weak WHEN conditions.
X14Cnm1	(ZEKE14C field name 1) Category: JCL Type: Global Reload requirement: `ZRELOAD GENOPTS` Required. Name of the JCL exit ZEKE14C's first field (up to eight alphanumeric characters long). This name will appear on the Event Master Record screen and is the keyword for the ZEKE batch utility program. The default value is `LIBNAME`. Do not use `PDS` or `MEMBER` as values for this field (these are predefined Zeke keywords).	
X14Cnm1L	(ZEKE14C field name 1 length) Category: JCL Type: Global Reload requirement: `ZRELOAD GENOPTS` Required. Specifies the length (*nn*) of the input field to be associated with the *name* field. The specified length is passed to ZEKE14C and can be up to 15 characters long (as long as the total length of both X14Cnm1L and X14Cnm2L does not exceed 16). The default value is `08`.	
X14Cnm2	(ZEKE14C field name 2) Category: JCL Type: Global Reload requirement: `ZRELOAD GENOPTS` Required. Name of the JCL exit ZEKE14C's second field (up to eight alphanumeric characters long). This name appears on the Event Master Record screen and is the keyword for the ZEKE batch utility program. Do not use `PDS` or `MEMBER` as values for this field (these are predefined Zeke keywords).	

Option	Description
X14Cnm2L	(ZEKE14C field name 2 length) Category: JCL Type: Global Reload requirement: `ZRELOAD GENOPTS` Required. Specifies the length (*nn*) of the input field to be associated with the *name* field. The specified length is passed to ZEKE14C and can be up to 15 characters long (as long as the total length of both X14Cnm1L and X14Cnm2L does not exceed 16). The default value is `08`. **Note:** A value for the X14Cnm2 field is not required in the EMR. If you do not specify a value for the X14Cnm2(L) generation option, the field value in the EMR will default to `MEMBER (` but you can clear it).
X14Coml	(ZEKE14*x* user exit storage) Category: Exits Type: Global Reload requirement: `ZKILL TRACK` Optional. 6-character numeric value that specifies the number of bytes of storage allocated and passed to the JCL user exits (ZEKE14*x*).
Zekectl	(ZEKECTL comment statements) z/OS only Category: JCL Type: Global Reload requirement: `ZRELOAD GENOPTS` Required. Indicates whether Zeke inserts //*ZEKECTL comment statements into the JCL when submitting a job. This option is required to enable Zeke's ThruPut Manager interface (see the appendix in the *ASG-Zeke Scheduling for z/OS User's Guide* for details). If you do not use ThruPut Manager, these comment statements still can be useful because they enable Zeke event information to be referenced in JES job logs. These are the valid values: `N` — Zeke does not insert //*ZEKECTL comment statements.

Option	Description	
	Y	Default. Zeke inserts //*ZEKECTL comment statements into the JCL when submitting a job. If the Posid generation option is set to Y, then the comment statements are inserted in front of the ZEKECTL POSID job step. The value of the Posidend generation option determines whether the comment statements are inserted at the beginning or end of the JCL.
ZPrdCom	(Z-product communications) z/OS only Category: General Type: Global Reload requirement: ZKILL TRACK Required. Indicates whether you want Zeke to communicate with other 'Z' products (e.g., Zebb). These are the valid values:	
	N	Do not activate interproduct communications.
	Y	Default. Activate interproduct communications to notify other Z-products (e.g., Zebb) of new or deleted Zeke events, or when a COPY or COPYALL function is performed in Zeke. **Note:** When ZPrdCom is set to Y, the load library for the other Z-product must be present in the Zeke started task, all ZEKE utility program jobs, and ISPF and online users (e.g., CICS, TSO, etc.).
ZPrdSEmr	(Suppress Z-product messages) z/OS only Category: General Type: Global Reload requirement: ZRELOAD GENOPTS Required. Indicates whether to suppress interproduct messages related to EMR processing. This option is valid only if ZPrdCom (page 606) is set to Y. These are the valid values:	
	N	Do not send messages related to EMR changes to other Z-products (e.g., Zebb); send only SQR messages.

Option	Description	
	Y	Default. Send both EMR and SQR messages.

A

Appendix A
XPath Location Paths

Records exported by the ZEKEXUTL import/export utility are written to the export file as XML elements and attributes. To reference individual XML data, the utility must navigate through the XML elements (nodes/objects) to locate a particular attribute, or property. XPath location paths enable this navigation ability.

General Format

An XPath location path looks similar to a UNIX or MS-DOS file system path name. Forward slash characters (/) separate the element names (like in a directory tree). For example:

```
"/name1/name2/@att"
```

where:

Element	Description
`/`	Indicates the root element.
`name1`	Indicates a child of the current element called `name1`.
`name2`	Indicates a child of the element called `name1`, called `name2`.
`@att1`	Indicates an attribute of the element `name2`, called `att1`.

Note:

These are the common abbreviations found in XPath location paths.

Every element has an *axis*, which indicates the relationship between consecutive elements. (See "Axis" on page 610 for a detailed explanation.) Any element name without an `"axis::"` defaults to an axis name of `"child::"`.

Expressed as a full XPath location path, the path in the previous example would include the axis for each element. For example:

```
"/child::name1/child::name2/attribute::att"
```

The two examples are identical paths.

When a path element does not have an axis rule, the axis defaults to `"child::"` with these exceptions:

Element	Description
`//`	All descendants of the context node, and the context node.
`.`	Context node (current element).
`..`	Parent node (parent of current element).
`@name`	Attribute of the context node with the specified name.
`/`	At the start of the path name, this indicates to make the root node the context node before applying the path name to the context node. In this utility, the starting context node is the *messageData* element, which is also the root. So, a leading `/` is not necessary.

Axis

An *axis* is the part of an XPath location path that describes the location of the next element in relation to the current, or *context*, element. The axis can be omitted and defaults to:

`"child::"`

This indicates that the next element is a child of the current element.

If an axis is specified in an XPath location path, it always appears before the element name. The axis name is delimited by two colon (:) characters that separate it from the element name. These are the axis names:

Axis Name	Description
`child::`	All children of the context node.
`descendant::`	All nodes contained inside the context node, that is, a child node, child of a child node, child of a child of a child node, and so on.

Note:

Only elements have child nodes. Attributes (properties) do not have children.

Axis Name	Description

`descendant-or-self::`	Any descendant of the context node or the context node itself.
	`//`
	is an abbreviation for
	`"/descendant-or-self::node()/"`

`parent::`	Element that immediately contains the context node. The context node is a child of this node.
	`".."`
	is an abbreviation for
	`"parent::node()"`

`ancestor::`	Root node and all element nodes that contain the context node.
`ancestor-or-self::`	All ancestors of the context node, as well as the context node itself.
`following::`	All nodes that begin after the context node ends, except for attributes.
`following-sibling::`	All nodes that follow the end of the context node and have the same parent node. **Note:** Attributes do not have siblings.
`preceding::`	All nodes that end before the context node begins, except for attributes.
`preceding-sibling::`	All nodes that end before the context node, except for attributes, and have the same parent node.

`attribute::`	All attributes of the context node.
	`"@name"`
	is an abbreviation for
	`"attribute::name"`

`self::`	Context node itself.
	`"."`
	is an abbreviation for
	`"self::node()`

Predicates

The ZEKEXUTL utility supports single-level predicates when using XPath path strings. A *predicate* is an expression within the square brackets embedded in the XPath path string. For example:

```
'//eventWhenList["predicate"]/eventWhen["predicate"]/@clauseText'="xxx"
```

The predicate must evaluate true to keep this statement in the result set.

Each node within the XPath path can have zero predicates or one predicate.

The predicate itself consists of XPath references to elements and attributes (the lterm), a relational operator, and a value that relates to the lterm (the rterm).

You must enclose the XPath path of the predicate in quotes, either single (') or double ("). You can also use doubled quotes if needed.

Example 1:

```
    CHANGE ALL

'//emrJobData["ancestor::*/event/@eventNum"="1"]/@initClass'="Q"
'//emrJobData["ancestor::*/event/@eventNum"!="1"]/@initClass'="A"
```

In this example, the 1st statement changes the `@initClass` attribute only if this element belongs to event number 1.

The second statement changes the `@initClass` attribute only if this element does not belong to event number 1.

Example 2:

These statements are equivalent, and show the various types of quotes that can be used:

Event (EMR) Field Names

These are the event (EMR) field names:

Field	XPath Location Path	Default Value
Acctg	"descendant::emrAccountingData/@dispatchCount"	
ACTive	"descendant::emrBaseData/@activated"	"1"

Field	XPath Location Path	Default Value
AEOE	"descendant::eventWhen/@clauseText"	"*AEOE *"
AEOJ	"descendant::eventWhen/@clauseText"	"*AEOJ *"
AEOP	"descendant::eventWhen/@clauseText"	"*AEOP *"
AEOS	"descendant::eventWhen/@clauseText"	"*AEOS *"
APPL	"descendant::emrBaseData/@appID"	**APPID**
AT	"descendant::eventWhen/@clauseText"	"*AT *"
BOJ	"descendant::eventWhen/@clauseText"	"*BOJ *"
BOP	"descendant::eventWhen/@clauseText"	"*BOP *"
CLass	"descendant::emrJobData/@initClass"	"*BOP *"
Cond	"descendant::conditionCodeList//@*"	"*"
DEACtivate	"descendant::emrBaseData/@activated"	"0"
DEScription	"descendant::emrBaseData/description/line/@*"	"*"
Doc	"descendant::emrDoc/*/line/@*"	"*"
DSn	"descendant::tapeDoc/tape/@name"	
EARlytime	"descendant::emrBaseData/@earlyTime"	
EMAILSUBject	"descendant::emailtask/@subject"	
EMAILTASK	"descendant::taskEventData/@taskType"	"2"
EMAILTO	"descendant::emailtask/@toaddr"	
ENAMe	"descendant::emrBaseData/@eventName"	
EOE	"descendant::eventWhen/@clauseText"	"*EOE *"
EOG	"descendant::eventWhen/@clauseText"	"*EOG *"
EOJ	"descendant::eventWhen/@clauseText"	"*EOJ *"
EOP	"descendant::eventWhen/@clauseText"	"*EOP *"
EOS	"descendant::eventWhen/@clauseText"	"*EOS *"

Field	XPath Location Path	Default Value
EVENTs	"descendant::event/@eventNum"	
EVTCal	"descendant::emrBaseData/@calendar"	
GROupid	"descendant::emrBaseData/@gID"	
HIGHIMPortance	"descendant::emailtask/@impint"	"1"
IMPORTAnce	"descendant::emailtask/@impint"	
IMPNAME	"descendant::emailtask/@importance"	
Jcl	"descendant::jobJCL/line/@text"	
JCLBook	"descendant::JCL/@book"	
JCLDDname	"descendant::JCL/@dd"	
JCLDSname	"descendant::JCL/@dsn"	
JCLFName	"descendant::JCL/@fname"	
JCLFType	"descendant::JCL/@ftype"	
JCLLibrary	"descendant::JCL/@library"	
JCLMember	"descendant::JCL/@member"	
JCLMOd	"descendant::JCL/@mod"	
JCLOpident	"descendant::JCL/@opident"	
JCLPAssword	"descendant::JCL/@password"	
JCLPRocedure	"descendant::JCL/@procedure"	
JCLPW	"descendant::JCL/@password"	
JCLSOurce	"descendant::JCL/@sourceName"	
JCLSTatus	"descendant::JCL/@status"	
JCLVersion	"descendant::JCL/@ver"	
JOB	"descendant::emrBaseData/@event.type"	"1"
JOBName	"descendant::emrJobData/@jobName"	

Field	XPath Location Path	Default Value
LATEEnd	"descendant::emrBaseData/@lateEnd"	
LATETime	"descendant::emrBaseData/@lateTime"	
LOGINName	"descendant::taskEventData/@loginName	
LOWIMPortance	"descendant::emailtask/@impint"	"3"
LVARName	"descendant::varData/@varName"	
LVARValue	"descendant::varData/@value"	
NORMALIMPortance	"descendant::emailtask/@impint"	"2"
Notes	"descendant::notePad/line/@text"	
NOTJ	"descendant::eventWhen/@clauseText"	"*NOTJ *"
NOTP	"descendant::eventWhen/@clauseText"	"*NOTP *"
Occurs	"descendant::emrOccurs/@text"	
OPErok	"descendant::emrBaseData/@needOpOk"	
Pad	"descendant::scratchPad/line/@text"	
PRIority	"descendant::emrBaseData/@dispatchPriority"	
Reply	"descendant::autoReply/@msgText"	"*"
RESOurce	"descendant::emrResource/@name"	"*"
REXX	"descendant::rexxEventData/@execName"	"*"
SCHENV	"descendant::emrBaseData/@schEnv"	
SCOM	"descendant::emrSCOMCommandData/@commandText"	"*"
SECGroup	"descendant::emrJobData/@securityGroup"	
SFTPHOST	"descendant::sftptask/@hostName"	
SFTPTASK	"descendant::taskEventData/@taskType"	"1"
STArt	"descendant::emrBaseData/@schedTime"	

Field	XPath Location Path	Default Value
SYStems	"descendant::emrBaseData/@sysID"	
TAPes	"descendant::emrJobData/@tapes"	
TARget	"descendant::emrBaseData/@target"	
TASKName	"descendant::taskEventData/@taskName"	
TASKType	"descendant::taskEventData/@taskType"	
Text	"descendant::textPad/line/@text"	
TIME	"descendant::emrBaseData/@schedTime"	
USERid	"descendant::emrBaseData/@userID"	
VAR	"descendant::eventWhen/@clauseText"	"*VAR *"
VCOm	"descendant::vcomEventData/emrText/@value"	
WEOE	"descendant::eventWhen/@clauseText"	"*WEOE *"
WEOG	"descendant::eventWhen/@clauseText"	"*WEOG *"
WEOJ	"descendant::eventWhen/@clauseText"	"*WEOJ *"
When	"descendant::eventWhen/@clauseText"	"*"
WORK	"descendant::wcEventData/emrText/@value"	
XEOE	"descendant::eventWhen/@clauseText"	"*XEOE *"
XEOJ	"descendant::eventWhen/@clauseText"	"*XEOJ *"
ZCOm	"descendant::zcomEventData/emrText/@value"	

Variable (VAR) Field Names

These are the variable (VAR) field names:

Field	XPath Location Path
APPL	"descendant::varData/@app"
DEScription	"descendant::varData/description/line/@text"
GROupid	"descendant::varData/@gID"
NAME	"descendant::varData/@varName"
NOTEpad	"descendant::varDoc/nodePad/line/@text"
PROCedure	"descendant::varVals/@proc"
SCRatchpad	"descendant::varDoc/scratchPad/line/@text"
SETter	"descendant::varVals/@setter"
SYStems	"descendant::varVals/@sys"
TEXTpad	"descendant::varDoc/textPad/line/@text"
USERid	"descendant::varData/@uID"
VALue	"descendant::varVals/@val"
VARname	"descendant::varData/@varName"

Calendar (CAL) Field Names

These are the calendar (CAL) field names:

Field	XPath Location Path
CALendar	"descendant::calendar/@name
NAME	"descendant::calendar/@name
YEAR	"descendant::calendar/@year
TYPE	"descendant::calBaseData/@calType

Field	XPath Location Path
WORKdays	"descendant::*/@workDays
FISCmnth	"descendant::calStd/@fiscalStartMonth
FISCMOnth	"descendant::calStd/@fiscalStartMonth
LASTdate	"descendant::calBaseData/@lastDate
STARTdate	"descendant::calBaseData/@startDate
ENDdate	"descendant::calBaseData/@endDate
EXPiredate	"descendant::calBaseData/@expireDate
SLACKdays	"descendant::calUser/@slackDays
HITdate	"descendant::calSpecHit/@hitDate
HOLidays	"descendant::*/@holidayDate
PERiod	"descendant::calPeriod/@periodDays
DOC	"descendant::calDoc/*/@text
SCRatchpad	"descendant::scratchPad/*/@text
NOTepad	"descendant::notePad/*/@text
TEXtpad	"descendant::textPad/*/@text

Login Field Names

These are the login field names:

Field	XPath Location Path	Default Value
ALLOW	"descendant::login_os/@allow_token_access"	
AUTH	"descendant::login_sftp/@auth_using_cert"	
CERTIficate	"descendant::login_sftp/@certificate"	
CERTPwd	"descendant::login_sftp/@certificate_pwd"	

Field	XPath Location Path	Default Value
COMPression	"descendant::login_sftp/@use_compression"	
DOMAIN	"descendant::login_os/@domain"	
HOSTName	"descendant::login_email/@host"	
HOSTName	"descendant::login_sftp/@host"	
HOSTType	"descendant::login_sftp/@host_type"	
MAC	"descendant::login_sftp/@host_type"	"3"
PASSIVE	"descendant::login_sftp/@passive"	
PORT	"descendant::login_sftp/@port"	
PROTocol	"descendant::login_sftp/@protocol"	
REQUIRESLOGin	"descendant::login_email/@requireslogin"	
SENDADDRess	"descendant::login_email/@sendaddress"	
SMTPPort	"descendant::login_email/@smtpport"	
UNIX	"descendant::login_sftp/@host_type"	"1"
WINdows	"descendant::login_sftp/@host_type"	"2"

Expressions

Expressions consist of these parameters:

'lterm' relop 'rterm'

where:

Parameter	Description

'lterm'	Keyword representing the field, for example:
	`EVENT`
	`SYSTEM`
	or an XML XPath location path name, for example:
	`"descendant::/event/@num"`
	`"descendant::/emrBaseData/@sysID"`
	XPath location paths must be enclosed in single or double quotes. Field names are not enclosed in quotes.

relop	Relational operator. These are the valid values: `EQ` `=` `==` Equal to `NE` `!=` `^=` Not equal to `GT` `>` Greater than `GE` `>=` Greater than or equal to `LT` `<` Less than `LE` `<=` Less than or equal to
'rterm'	Numerical value, word, or string. If the value contains spaces or punctuation characters, or starts with a numeric character, then you must surround the value in either single or double quotes. Otherwise, you do not need to surround the value with quotes. If you are unsure whether values to be changed might contain these characters, ASG recommends that you use either single or double quotes.

Syntax Rules

These are the rules for formatting expressions:

- If the value contains spaces or punctuation characters (or starts with a numeric character), then you must surround the value in either single or double quotes. Otherwise, you do not need to surround the value with quotes. If you are unsure whether values to be changed might contain these characters, ASG recommends that you use either single or double quotes.
- Use parentheses to group compound expressions together so that they will be evaluated in the correct context.
- You can construct a compound expression by including `AND` or `OR` between expressions (including expressions nested within parentheses).

Appendix B
LE Runtime Options

This appendix describes how to override the Language Environment (LE) program runtime options to customize your environment.

Caution! ASG recommends that you do no override the LE program runtime options unless you have extensive LE experience/knowledge.

Language Environment (LE) Overview

The Language Environment (LE) is an element of z/OS that provides a common runtime environment and common runtime services for applications.

Within a Language Environment, these are the three levels of LE runtime options:

Level	Option Type	Description
1	Installation Runtime Options	These are the system-wide defaults for all programs.
2	Program Runtime Options	These are the ASG-defined defaults that have been compiled into ASG programs. These options override the Installation Runtime Options.
3	Override Runtime Options	These options override the Program Runtime Options. See the following sections for details.

Overriding the Program Runtime Options

The LE options specified in ASG program modules provide reasonable and predictable behavior for most, if not all, customers. If necessary, you can override the program runtime options. This enables you to customize the Language Environment for the Zeke server or the Import/Export utility if that the default LE options do not provide the desired environment. Overriding the LE options can also assist ASG in resolving support issues; ASG Customer Support might ask you to change one or more LE options to provide for better dump captures. ASG also might recommend other adjustments to tune the performance of the programs in your environment.

Caution! ASG recommends that you do not override the LE program runtime options unless you have extensive LE experience/knowledge. Changing the LE options can have unpredictable and/or harmful results and can cause an affected program to become unstable.

Zeke Server (ZEKE6SRV)

You can override the LE options for the Zeke server (ZEKE6SRV) using an LEPARMS dataset allocated to the Zeke address space. When Zeke starts, it reads the LEPARMS file and stores the override options in memory. The options then are passed to ZEKE6SRV when it is attached by Zeke.

Any LE override options must exist in a dataset allocated to the LEPARMS DD name in the Zeke address space started task:

```
//LEPARMS DD DISP=SHR,DSN=hlq.prod.leparms
```

The LE parameters dataset can be any sequential dataset, variable length, any LRECL that you can edit. The file can contain comment lines indicated by an asterisk (*) in the first column of the line. Place each LE option on a separate line, with no line number.

Figure 1 illustrates a sample LEPARMS File:

Figure 1 • Sample LEPARMS File

```
* This file contains LE parameters that will be supplied
* to the ZEKE6SRV module when it is started.
*
* Any line starting with '*' is treated as a comment by
* the LE options loader.
RPTOPTS(ON)
RPTSTG(ON)
*HEAPCHK(ON,1,0,0)
ANYHEAP(1M,1M,ANY,FREE)
HEAP(6M,6M,ANY,FREE,1K,1K)
```

Note:

See also "Tuning Notes and Precautions" on page 626 before changing the LE options.

Import/Export Utility (ZEKEXUTL)

You can override the LE options for the Import/Export utility (ZEKEXUTL) by placing the options in the PARM='...' parameter of the EXEC statement for the job step that executes the ZEKEXUTL program.

The LE options must appear *before* the other parameters that are passed to the program. A forward slash (/) must separate the LE options from the program options specified in the PARM='...' of the EXEC statement.

For example:

```
//MYSTEP01 EXEC PGM=ZEKEXUTL,REGION=0M,
//  PARM='RPTOPTS(ON),RPTSTG(ON)/SUBSYS=SSSI'
```

Note:

See also "Tuning Notes and Precautions" on page 626 before changing the LE options.

Zeke Address Space Commands

Zeke provides its own operator modify commands for altering the Zeke environment. The Zeke main program (ZEKE47A), which manages the Zeke environment, is called by SSS4001 (the main program module for the Zeke started task). The Zeke started task creates the subtasks that execute in the address space. Each subtask supports a key Zeke feature and runs independently of other subtasks. Generally, the Zeke main task manages all subtasks, but you might occasionally need to intervene manually.

Note:

When the Zeke started task starts the Zeke server subtask (ZEKE6SRV), only the subsystem name is included as a parameter.

By default, Language Environment (LE) runtime options are set for a default LE environment. You can update the runtime options in memory for the Zeke server subtask by issuing a modify command to the Zeke address space. The command prefix is #.

`F xxxZ620A,#command`

Or

`MODIFY xxxZ620A,#command`

Address space commands are operator commands than can be used to change the in-memory LE runtime options for a subtask. These changes persist only until the address space is restarted. Address space commands do not change the runtime options for an active subtask. For the changes to take effect, the subtask must be restarted.

Address space commands can be issued against all subtasks, unless noted otherwise.

See "Zeke Address Space Commands" on page 401 for commands, parameters, and syntax, and output samples.

See Appendix C, "Other Address Space Commands," on page 629 for additional commands that are supported by the ZEKE6SRV module only.

Tuning Notes and Precautions

Keep these considerations in mind if you decide to alter the LE options:

- Because the Zeke server and ZEKEXUTL (import/export) utility are thread-based and use 31-bit addressing, the STACK should always be above the 16M line. For example:

  ```
  STACK(,,ANY,KEEP,,) and ALL31(ON)
  ```

- In the event of an out-of-storage condition, LE must have a recovery segment to enable processing of the condition. If a recovery segment is not available, LE will issue a U4088 abend and terminate the thread. To avoid this situation, set the last parameter of the `STORAGE` option to 64K or more. For example:

  ```
  STORAGE(NONE,00,00,64K)
  ```

- Threads obtain buffers from the LE heap. For LE to perform well, heap segments should be above the 16M line and as large as is practical. For example:

  ```
  HEAP(18M,6M,ANY,KEEP,1K,1K)
  ```

- When it comes to heap storage, the fewer segments the better. If possible, increase the initial heap segment size to reduce the number of heap segments that must be scanned to satisfy requests from the threads. If you need 100M of heap storage, better performance will be achieved with one 100M heap segment than with a hundred 1M heap segments.

- Performance is generally better when `HEAPPOOLS(ON,...)` is enabled for small memory allocations. When tuning heap performance you should use `RPTOPTS(ON)` and `RPTSTG(ON)`, and review the output generated when the Zeke server is shutdown. The storage report created to the SYSOUT DD name will provide some insight into the heap pool performance. The report is near the end of the SYSOUT output.

Note:

The Zeke server has these default LE settings (which perform well for most users):

```
STACK(128K,128K,ANY,KEEP,512K,128K)
STORAGE(NONE,FA,00,64K)
HEAP(18M,6M,ANY,KEEP,1K,1K)
HEAPPOOLS(OFF)
```

The critical values shown are the minimum recommended values, which you can increase, if needed. You also can enable HEAPPOOLS(ON,...) to gain better performance for small memory allocations. The actual settings for HEAPPOOLS(ON,...) will depend on your needs as determined by the storage report generated by the RPTSTG(ON) LE runtime option.

C Appendix C
Other Address Space Commands

This appendix provides the address space commands that can be issued against the ZEKE6SRV subtask. (These commands cannot be issued against any other subtask.)

Caution! Use these commands only at the direction of ASG Customer Support.

Enabling the Commands

To be able to use the commands to start or restart ZEKE6SRV, you must first copy the RI Server environment file ENVIRON to a temporary dataset and use the temporary dataset in the ZEKE step in the Zeke started task.

- Update your Zeke started task by adding a step to copy the RI Server environment file ENVIRON to a temporary dataset.

```
//*
//*   ZEKE  :  STARTED TASK PROC
//*
//ASGZR620A PROC XPROC=ASGX320A,R=0K,S=ZK62,ZREL=R620A,
//             XREL=R320A,OASIS=(53,L),RREL=R212A
//*
//ENVIRON EXEC PGM=IEBGENER
//SYSPRINT  DD SYSOUT=X
//SYSUT2    DD DISP=(NEW,PASS),DSN=&&ENVIRON
//SYSUT1    DD DISP=SHR,DSN=ZEKE.&ZREL..ENVIRON,
//             FREE=CLOSE
//SYSIN     DD DUMMY
//*
//ZEKE EXEC PGM=SSS4001,REGION=&R,TIME=1440,ACCT=10066,
//PARM='OASIS=&OASIS,ZEKE=(53,L),SUBSYS=&S,XPROC=&XPROC,END'
//*
...
//ENVIRON   DD DISP=OLD,DSN=&&ENVIRON
...
```

Caution! If you want to be able to restart the ZEKE6SRV subtask, do not use FREE=CLOSE on the ENVIRON DD. This prevents the ENVIRON file on the ZEKE step from being released.

Command Format

You issue address space commands as a modify command to the Zeke address space. The command prefix is #. For example:

```
F xxxZ620A,#
```

See "Zeke Address Space Commands" on page 401 for more information.

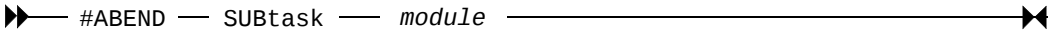
#ABEND Command

Caution! Use this command only at the direction of ASG Customer Support.

Use this command to terminate a subtask module (with completion code 13E).

The subtask is not automatically restarted. The subtask remains stopped until it is manually started or the Zeke address space is restarted.

Syntax



Parameters

These are the valid parameters for the #ABEND command:

Parameter	Description
SUBtask	Terminate the specified subtask.
module	Name of the subtask to be terminated. For example: `F xxxZ620A,#ABEND SUB ZEKE6SRV`

Sample Output

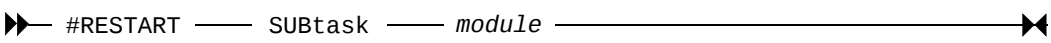
```
Z470LI #ABEND SUBTASK ZEKE6SRV
BPXP018I THREAD 248B405000000007, IN PROCESS 83886164, ENDED
WITHOUT BEING UNDUBBED WITH COMPLETION CODE 0013E000,
AND REASON CODE 00000000.
BPXP018I THREAD 248A715000000006, IN PROCESS 83886164, ENDED
WITHOUT BEING UNDUBBED WITH COMPLETION CODE 0013E000,
AND REASON CODE 00000000.
BPXP018I THREAD 248A23B000000005, IN PROCESS 83886164, ENDED
WITHOUT BEING UNDUBBED WITH COMPLETION CODE 0013E000,
AND REASON CODE 00000000.
BPXP018I THREAD 248A16C000000004, IN PROCESS 83886164, ENDED
WITHOUT BEING UNDUBBED WITH COMPLETION CODE 0013E000,
AND REASON CODE 00000000.
BPXP018I THREAD 248A09D000000003, IN PROCESS 83886164, ENDED
WITHOUT BEING UNDUBBED WITH COMPLETION CODE 0013E000,
AND REASON CODE 00000000.
BPXP018I THREAD 2489A25000000002, IN PROCESS 83886164, ENDED
WITHOUT BEING UNDUBBED WITH COMPLETION CODE 0013E000,
AND REASON CODE 00000000.
BPXP018I THREAD 248954B000000001, IN PROCESS 83886164, ENDED
WITHOUT BEING UNDUBBED WITH COMPLETION CODE 0013E000,
AND REASON CODE 00000000.
Z470KI Zeke subtask ZEKE6SRV State STOPPED  Abends 001 Max 020
```

#RESTART Command

Caution! Use this command only at the direction of ASG Customer Support.

Use this command to stop and restart a subtask module without having to restart the Zeke started task. This command essentially stops the subtask, waits for it to terminate, deletes the subtask, and attaches the new subtask.

Syntax



Parameters

These are the valid parameters for the #RESTART command:

Parameter	Description
`SUBtask`	Restart the specified subtask.
`module`	Name of the subtask to be restarted. For example: `F xxxZ620A,#RESTART SUB ZEKE6SRV`

Sample Output

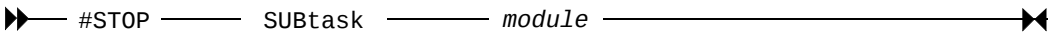
```
Z470LI #RESTART SUBTASK ZEKE6SRV
Z470KI Zeke subtask ZEKE6SRV State RESTART  Abends 000 Max 020
Z6006I ZEKE OpsCentral Server quiesced
Z6004I ZEKE OpsCentral Server terminating
Z470SI ATT:RPTOPTS(ON),RPTSTG(ON),HEAPCHK(ON,1,0,0)
Z0725I Zeke Subtask ZEKE6SRV has been reattached, 020 Remaining
Z6003I ZEKE OpsCentral Server enabled
```

#STOP Command

Caution! Use this command only at the direction of ASG Customer Support.

Use this command to stop a subtask module.

Syntax



Parameters

These are the valid parameters for the #STOP command:

Parameter	Description
`SUBtask`	Stop the specified subtask.
`module`	Name of the subtask to be stopped. For example: `F xxxZ620A,#STOPPED SUB ZEKE6SRV`

Sample Output

```
Z470LI #STOP SUBTASK ZEKE6SRV
Z470KI Zeke subtask ZEKE6SRV State STOPPING Abends 000 Max 020
Z6006I ZEKE OpsCentral Server quiesced
Z6004I ZEKE OpsCentral Server terminating
Z470KI Zeke subtask ZEKE6SRV State STOPPED  Abends 000 Max 020
```

Index

Symbols

#ABEND command 630
#CLEAR command 403
#DISPLAY command 404
#RESET command 405
#RESTART command 632
#SET command 406
#START command 407
#STOP command 633
$CLOSE command 408
$DISPLAY command 410
$JOBEXIT 552
$KILL command 416
$OPEN command 419
$SET command 420
$XXXXX special name 174

A

ABCODE special name 173
abend jobs, number of days to retain 575
ABTYPE special name 173
address space commands 401–407, 625, 629–633
alerts 45–46
  closing 408
  opening 419
allocating XML datasets 116
application ID 35
  event 428
  variable 428
audit
  calendar 430
  generation options 457
  initiator 485
  name and address 470
  partition 485
  pool changes 488
  resource changes 490
  security class 494
  security operator 494
  SQRs 492
  status changes 462
  variables 517
  Zeke commands 517
auto replies, enabling 528
automatic restarts for a subtask 406
average duration 37, 429

B

backing up the Zeke database 8–13
  database backup size report 10
  to a data space 9
  to disk 9
  to tape 9
batch utility program
  BACKUP command 8–13
    examples 10
  CALENDAR command 14–18
  CLEARCPU command 19
  CREATE command 20–21
  DOC command 23–28
  EVENT command 29–61
  formatting rules 3–7
  GENOPTS command 68–69
  OPTION function 78
  OVERRIDE command 102–104
  RECOVER command 83
  REPORT command 104–106, 111
  RESOURCEE command 85–86
  RESTORE command 87–92
  SCHEDULE command 93–106
  SIMULATE command 106
  SIMULATE function 107
  VAULT function 113
Bim-Edit 24, 45, 70
  options 529

C

calculating dates (CDATE) 186
calendar
  fiscal start 455
  holidays 458
  periods 486
  slack days (in a calendar) 473
  special days 436
  special months 461
  year 516
CALENDAR batch function 14–18
  examples 17
calendars 37
  adding and maintaining 14–18
  deleting 15, 22, 72
  expiration date 15
  fiscal month 15
  holidays 16

in use by EMRs 16
maintaining 14
periods 16
slack days 16
special 16
workdays 16
year 17
CAPS mode 507
catalog, *see* database
CCACTION 52
CDATE, date calculations 186
changing values in imported/exported records 144–151
class
dispatch 440
job 601
CLEARCPU batch function 19
clearing LE parameters 403
closing an alert 408
CMS
files 47
machine name 601
option 532
wait for JCL 602
code, *see* condition code
command events 33, 60
commands
address space 401–407, 625, 629–633
#ABEND 630
#CLEAR 403
#DISPLAY 404
#RESET 405
#RESTART 632
#SET 406
#START 407
#STOP 633
issuing via SET xCOM 176
OpsCentral server 408–423
$CLOSE 408
$DISPLAY 410
$KILL 416
$OPEN 419
$SET 420
see also batch utility program, operator commands
completed events, retaining 575
COMRC special name 174
condition code
action 425
CCACTION 52
defining for an event 52–55, 59
low range 466
maximum 442
maximum at EOJ 466
operator 483
procedure step 489
processing 52–55
range 457
setting codes using ZEKESET 181
step name 498
Condor 24, 46, 70, 533
console alarm
late events 552
message events 556
control statements, import/export 122–126
conventions page x
copying
documentation to the Zeke database 23–28
events 30
JCL to the database 70–71
CPUID special name 174
CREATE batch function 20–21
sample JCL 21
creating
a sequential disk file 203, 221
the Zeke database 20–21

D

data space
for database backup 9
use in report generation 202, 215, 235, 243, 245, 247, 249
database
backup 8–13
to a data space 9
to disk 9
to tape 9
block size 21, 89
changing values in imported/exported records 144–151
copying documentation into 23–28
creation 20–21
database backup size report 10
exporting records to a file 127
filtering imported/exported records 138–142
format
logical 8–9
physical 8–9
importing records from a file 134
maximum number of events 89
percentage used 564
restoring the database 87
status 11
status report 78, 80
vault, disabling 113
Database Backup Size Report 11

Database Status Report 78
dataset
    default PDS 565
    documentation 25, 27
    media 500
    triggering hold code 324
    type 458
    version number 509
    volume information 510
    XML, allocating 116
date calculations (CDATE) 186
DATE special name 174
DATEJ special name 174
DAY special name 174
DD name for PDS 486
deactivating events 30
deleting 30
    calendars 15, 22, 72
    corrupted events forcibly 31
    documentation from an event 24
    events 30
    JCL from an event 70
deleting a corrupted event forcibly 11
deleting forcibly 31
delimiters
    operator commands 258
dependencies 45
disaster recovery level 38, 440
dispatching
    delay for pooled events 538
    frequency 47
    initiators/partitions selection 538
    keywords 511
    late events 568
    number of times 504
    priority 38, 440, 534
    recurring events 504
    remove events from queue 570
    retained events 575
    seconds to wait 538, 557
    wait for operator OK 483, 561
    wait for pending event 565
    when EOJ 546
dispatching prerequisites 45
display
    GENOPTs ??–349
    operator OK 470
DOC batch function 23–28
    sample JCL 27
documentation
    dataset 25, 27
    event 24
    note 25, 27, 472
    scratch 25, 27, 493
    sources 24
    tape 501
    text 25, 503
download status 439
DSN, multiple events with same 541
duration, *see* average duration

E

early time 38
EDB index 545
electronic vaulting, *see* vaulting
EVENT batch function 29–61
events 30–31
    activating 31
    adding documentation 24
    adding JCL 70
    adding same name 558–559
    adding to the schedule 29, 484
    command 33, 60
    copying 30
    deactivating 30
    defining condition codes 52–55, 59
    defining resources 55–59
    deleting corrupted events forcibly 11
    deleting documentation from 24
    deleting JCL 70
    dispatching
        prerequisites 45
    documentation sources 24
    EVENT special name 174
    expiration 38
    frequency 39
    job events 32–51
    listing events 213
    maintaining 29–61
    maximum number in database 21
    message 33, 60
    milestone 40
    percentage complete 425
    permanent 41, 44, 209–210, 292, 294, 486
        deleting 300
        reporting on 221
        scheduling 274
    platform 46
    reactivating 31
    reason codes 443
    recurring 43, 47
    refreshing events 283, 383
    replacing documentation 24
    replacing JCL 70
    retaining 42
    retaining completed 575
    REXX 34

scheduled multiple times 557
scheduling multiple times 469
status codes 324, 330, 443, 498
templates 36, 503, 508
triggers 43
types 443
updating 31
versions 44, 509
adding to the schedule (batch) 44
maintaining versions using operator commands 272
VM CP command 34
work center 35, 43
Zeke command 34
Zeke-controlled 37
executing the ZEKEXUTL (import/export utility) 120–126
expiration
calendar 15
event 38
exporting database records 116–166
changing record values 144–151
export reports 152–159
filtering 138–142
to a file 127
extended dependencies, *see under* WHEN conditions
External Security Interface (ESI) 546

F

FBA devices 548
FBOK status 445
filtering imported/exported records 138–142
fiscal months (in a calendar) 15
forecasting the schedule 94
frequency 47
calculations 39
FSucc status 456
function codes, *see* condition codes

G

generation options
maintaining 68–69
reloading 389
GENOPTs
displaying 349–350
global 91
GENOPTS batch function 68–69
global GENOPT 91
GRESOURCE parameter 79
GRESSYID parameter 79
group
replacing on JOB card 571
group ID 36

H

HIGHCOND special name 174
holidays (in a calendar) 16

I

ICCF 548
IDCAMS 549
IF clauses within SET statements 172
importing database records 116–166
changing record values 144–151
filtering 138–142
from a file 134
import reports 152–159
initiators 485
selection of 538

J

JCL 47, 51
adding to an event 70
copying to the database batch utility program) JCL command 70–71
target system 501
Zeke job control statements 169
JCL batch function 70–71
JCL libraries
Bim-Edit 529
CMS 532
Condor 533
default PDS 565
default source 535
EMR 462
EMR JCL options 551
JCL exit 604
JCL sources with POSID 566
Librarian 547, 552
OWL 562
Panvalet 563
Vollie 602
Zeke 462
JCL source 32, 70
JES 48
job events 32–51
job log 565
job priority, default 536
jobname
JOBNAME special name 174
replacing on JOB card 572
with ZDISPLAY 268

L
LASTCOND special name 174
LASTEVENT batch parameter 24, 35, 70
LASTPGM special name 174
LASTSTEP special name 174
late end time 39
late events priority 568
late start time 39
LE parameters 406
  clearing 403
  displaying 404
  setting 406
Librarian 25, 48, 71, 547, 552
LITERAL special name 175

M
message events 33, 60
messages
  dispatch 483
  number of minutes between events 565
  variables 531
milestone events 40
multiple event versions 567
must end time 40

N
NAME1 parameter 79
NAME2 parameter 79
NAME3 parameter 79
NAME4 parameter 79
NETREGID 289, 561
NNN special name 175
nonworking days 41
not after time 40
NOT-CAT2 messages, ignore 549
NOTDURING processing
  displaying wait details 317
  purging an job 376
  removing an inactive job 388
NOTDURING waits 331
note documentation 25, 27
notepad 36
NOTRIG option 292

O
OASIS variables 496
OCCURS clauses 41
  list of keywords 474
  options for non-workdays 561
ON EOM command (end-of-month rounding) 194
opening an alert 419
operating system
  default 537
  specify platform 487
operator commands
  abbreviations 256
  delimiters 258
  entering commands from console, /ZC, or ZCOM 253
  multiple parameters 257
  ZADD 274
  ZALTER 286
  ZDISABLE 308
  ZDISPLAY 311
  ZENABLE 360
  ZID 365
  ZINFO 366
  ZKILL 367
  ZMAP 370
  ZOK 372
  ZREFRESH 383
  ZRELEASE 384
  ZRELOAD 388
  ZRESOURCE 391
  ZSCAN 395
  ZSET 397
  ZSTATUS 400
operator IDs
  mixed case IDs 507
  required 574
  VM 602
operator OK 41, 561
operators in IF clauses 172
OpsCentral server
  commands 408–423
  displaying statistics 410
  setting threads or trace flags 420
  statistics 410
  terminating sessions or threads 416
  threads or trace flags 420
OPTION batch function 78
OSCAR, *see* OASIS
OUTFILE parameter 203, 221
output reports to file 203
OVERRIDE batch utility 102–104
OWL 562

P
Panvalet 25, 71, 563
partition
  number of dynamic partitions 436
  pause 564
  selection 538
  type 505
partitioned dataset (PDS) 25

path, level 466
PDS 49
member name 467
percentage complete (event) 425
periods (in a calendar) 16
permanent events 41, 44, 209–210, 292, 294, 486
deleting 300
reporting on 221
scheduling 274
permanent XML datasets 118
platform 46, 49, 487
POWER
bypass job cards 530
commands 485
display JCL on console 565
job cards 600
jobname 487
reader exit name 569
scan for class 571
subtasks and SCOMs 576
user 489
predicates, XPath 612
prerequisites, *see* WHEN conditions
printing the database status report 78
priority
dispatch 38
job, default 536
late events 568
operating system 50
ranges, job 536
problem tracking facility 564
procedure name, start rerun 497

R

RACF surrogate processing 571
RDATE 93
reactivating events 31
reader exit name 569
reason codes 324, 330, 443
RECOVER batch function 83
recurring events 43, 47, 504
dispatch time 456
number of times to dispatch 504
refreshing events 283, 383, 490
reloading generation options 389
remote job triggers 570
REPORT batch utility 104–106, 111
report writer facility
calendar listings
sample reports 235, 240
creating a sequential disk file 203, 221
event master listings
fields parameters 208
line spacing parameters 206
page break parameters 205
reporting parameters 213
sample reports 377–378
sort sequence parameters 204
formatting reports 201
output report to file 203
printing the database status report 78
report types 200
schedule listings
fields parameters 208
line spacing parameters 206
page break parameters 205
reporting parameters 213
sample reports 231–232
sort sequence parameters 204
variable listing
sample reports 244, 246–247, 251–252
reports
company name and address 79
database backup size 10
database status 79–80
printing company name and address 78
system table status 81
reports, import/export 152–159
rerun jobs 590
resetting the subtask restart counter 405
RESOURCE batch function 85–86
resources
amount required 433
audit 490
available 426
defining for an event 55–59
for restarts 425
holding 457
maintaining 85–86
modes 467
names 490
number shared 497
release 441
scope 493
status 497
restart
counter, resetting the subtask 405
reason 432
restarting a subtask module 632
restoring the database, RESTORE batch utility 87–92
retaining
completed events 575
events 42

return codes, *see* condition codes
REXX events 34
route command responses, option to 532
routing messages 556

S

SCHDT keyword 266, 282
schedule
  adding events
    options 484
  date 458
  for non-workdays 473
  forecasting 94
  holidays 561
  keywords 474
  listings 213
  SCHEDULE batch utility 93–106
    OVERRIDE command 102–104
    REPORT command 104–106, 111
    simulating the schedule 94
  simulation 94, 106
  storage for records 499
  storage statistics 366
schedule time 42
Schedule View
  date format 436
  sorting data 426, 455, 471
scheduling environments 42
SCOM events 33
  command type 434
  minutes until times out 576
scratch documentation 25, 27
search criteria 495
security
  access denied 577
  access for online functions 427
  batch verification 577
  element length 465
  external class 444
  external security
    elements 445
    security class 459
  External Security Interface 546
  field description 445
  how to handle access requests 488
  how to verify 578
  resource type 459
  starting element 497
  storage for exit routine 577
  system name 435
  user ID 507
  Zeke 482
security group 50
selection criteria
  save permanently 487
separator pages, creating 175
sequential disk file, creating 203, 221
server
  commands 408–423
  OpsCentral, displaying statistics 410
SET clause, IF clause within 172
SET ROUTE statement 175
SET VARIABLE statement 179
SET xCOM, issuing system and Zeke commands 176
SIMULATE batch function 106
simulating the schedule 94, 106–107
Simuload 94
slack days 473
slack days (in a calendar) 16
SMF
  IEFU83 exit 549
  option to install exits 545
  suppress records 601
special calendars 16
STALEA download status 294
STALEZ download status 293
starting a subtask module 407
status
  ABOK 425
  AEOJ 444
  audit changes 462
  database 80
  database status report 78
  download 439
    STALEA 294
    STALEZ 293
  event status 323–324, 330, 443
  Fail 444
  FBOK 445
  force to success 455
  FSucc 456
  not dispatched 470
  pending 486
  production 498
  reason codes 443
  restart 498
  work center 498
status codes, last execution 498
STATUS parameter 79
step name
  start rerun 497
  whether to log Zeke 580
stopping a subtask module 633
storage tables 366
subtasks

displaying information about 404
for SCOMs 576
resetting the restart counter 405
restarting 632
setting automatic restarts 406
starting 407
stopping 633
terminating 630
suppressing trace output from Zeke server logs 422
surrogate processing 571
SYSIN 26, 28
system
command events, *see* SCOM events
default ID 537
groups of systems 488
hold after start up 562
type 484
system ID 37
System Table Status Report 81

T

tape drives 51
calculating 530
number of 472
number required by job 501
start I/O 582
target system
for JCL 501
job execution 43
template
job default 582
message default 582
PCOM default 583
REXX default 583
SCOM default 583
using event templates 503, 508
VCOM default 584
work default 584
ZCOM default 584
templates, event 36
temporary XML datasets 118
terminating
a subtask module 630
OpsCentral server sessions or threads 416
text documentation 25
THISPGM special name 174
THISSTEP special name 174
time
early 38
late end 39
late start 39
must end 40
not after 40
schedule 42
TIME special name 174
trace messages
displaying 351
specifying messages to be generated 421
suppressing from Zeke server logs 422
Zeke server 122
triggering events 43
hold code 324

U

uncompleted events, retain 576
UNIX 50
updating event definitions 31
user IDs 37
blanking out an incorrect user ID in an EMR 37
mixed case IDs 507
replacing on JOB card 573
VM 602

V

VAR and ?VAR keywords 496
variables 43
current value 436
defining a new variable 179
how value was set 508
message 531
OASIS 496
previous value 488
SET clause 496
setting values 179, 397
setting with VAR and ?VAR 496
substitution 580
type 505
user ID 508
vaulting, VAULT batch utility 113
VERSION special name 174
versions 44, 567
see also event versions
virtual memory 510
VM
commands 509
operator ID 602
user ID 602
VM CP command events 34
VMEM, calculate 530
Vollie JCL library 602

W
wait time 455
WEAK conditions 292, 515, 603
WHEN conditions 45
  extended dependencies 515
  list of keywords 511
  listing events with a particular WHEN condition 225
  OCCHIT screen 442
  storage for 499
  WEAK conditions 292, 515, 603
work center events 35, 43
  comment lines 435
  loading into SQT storage 554
  retaining in the schedule 532
  SET clause 496
  user ID 508
workdays (in a calendar) 16

X
X14Cnm1 generation option 51
X14Cnm2 generation option 51
XEOJ/XEOE keywords 515
XML
  datasets
    allocating 116
    temporary and permanent 118
  elements and attributes 609–621
  paths 125
XPath
  fields 125
  location paths 609–621
  predicates 612
XVAR and ?XVAR keywords 496
XXX special name 175

Y
year (calendar) 17

Z
ZADD operator command 274
ZALTER operator command 286
ZDISABLE operator command 308
ZDISPLAY operator command 311
Zeke command events 34
Zeke server
  logs, suppressing trace output 422
  trace messages 122
Zeke-controlled events 37, 435
ZEKECPU special name 174
ZEKESET function
  branching within control statements 183, 185
  creating large block letter separator pages 175
  date calculations 186
  issuing CP commands 179
  SET statement syntax 170
  setting a wait interval 184
  setting condition codes 181
  setting return codes 181
  special name values 173
  terminating a job 175
ZEKEXUTL (import/export utility) 116–166
  control statements 122–126
  executing 120–126
ZENABLE operator command 360
ZID operator command 365
ZINFO operator command 366
ZKILL operator command 367
ZMAP operator command 370
ZOK operator command 372
ZREFRESH operator command 383
ZRELEASE operator command 384
ZRELOAD operator command 388
ZRESOURCE operator command 391
ZSCAN operator command 395
ZSET operator command 397
ZSTATUS operator command 400