About the Command Logs

Two command log layout formats are supported by Adabas mainframe products:

- The first format (invoked by CLOGLAYOUT=5) is supported in all supported Adabas mainframe versions and is the default format.

  The DSECT that maps the layout of the CLOGLAYOUT=5 records is called LOREC and can be found in the ADA*vrs*.SRCE library.

- The second format (invoked by CLOGLAYOUT=8) is supported only in Adabas version 8 or later. This format includes all of the information provided in CLOGLAYOUT=5, but each buffer is prefixed by its corresponding Adabas buffer description (ABD) as well. Each segmented buffer (format, record, or multifetch) is written separately and uniquely identified.

  The DSECT that maps the layout of the CLOGLAYOUT=8 records is called LORECX and can be found in the ADA*vrs*.SRCE library.

This document describes the command log formats that Adabas supports as well as some command log utilities.

- CLOGLAYOUT=5 Record Layout
- CLOGLAYOUT=8 Record Layout
- Command Type Field in CLOGLAYOUT Records
- Remote Nucleus ID Field in CLOGLAYOUT Records
- Command-Executing Nucleus ID Field in CLOGLAYOUT Records
- Timestamps in CLOGLAYOUT Records
- PRILOG : Printing the Command Log
- PRILOGC: Printing the Command Log

Notation *vrs*, *vr*, or *v*: When used in this documentation, the notation *vrs* or *vr* stands for the relevant version of a product. For further information on product versions, see *version* in the *Glossary*.

CLOGLAYOUT=5 Record Layout

Command log layout 5 comprises two log record types:

- the basic log record type (x’0001’) is produced for all commands processed on noncluster nuclei and for those that arrive from a remote nucleus and run under internal command queue elements (ICQEs).
- the asynchronous request log record type (x’0002’) is created on a nucleus that sends a command to another nucleus. This record type is used in Adabas nucleus cluster environments only.

The DSECT that maps the layout of the CLOGLAYOUT=5 records is called LOREC and can be found in the ADA*vrs*.SRCE library.

CLOGLAYOUT=8 Record Layout

Command log layout 8 comprises two log record types:

- the basic log record type (x’0001’) is produced for all commands processed on noncluster nuclei and for those that arrive from a remote nucleus and run under internal command queue elements (ICQEs).
- the asynchronous request log record type (x’0002’) is created on a nucleus that sends a command to another nucleus. This record type is used in Adabas nucleus cluster environments only.

The DSECT that maps the layout of the CLOGLAYOUT=8 records is called LORECX and can be found in the ADA*vrs*.SRCE library.

Command Type Field in CLOGLAYOUT Records

The command type field (at hexadecimal offset 12 in the LOREC DSECT and hexadecimal offset 23 in the LORECX DSECT) refers to the following command types:

Update	A1/A4, E1/E4, N1/N2
Simple	All non-update commands with a single search argument.
Complex	All non-update commands with more than one search argument.

The command type field may contain other information that is unrelated to commands. The flags should therefore be tested with a binary mask:

```
B’xxxx xusc’
```

where *u* is the update flag bit, *s* is the simple flag bit, and *c* is the complex flag bit.

Remote Nucleus ID Field in CLOGLAYOUT Records

The remote nucleus field (at offset 4D in the LOREC DSECT and offset 1C in the LORECX DSECT)represents a remote nucleus ID.

- For the basic log record type (x’0001’) where the command originates in another nucleus, this field contains the ID of the nucleus that sent the command. It remains zero for locally-executed commands and noncluster nuclei.
- For the asynchronous request log record type (x’0002’), this field contains the ID of the nucleus to which the command is sent. It is zero for implicit broadcasts (GLOBAL and SYSTEM) and contains the ID of the first destination for explicit broadcasts (list of destinations). The CLOG record contains the Adabas control block, buffers, and (if appropriate) Adabas buffer descriptions (ABDs) related to the first (or only) destination, and the job name and communications ID of the initiator of the request.

Command-Executing Nucleus ID Field in CLOGLAYOUT Records

The nucleus ID field at offset 58 in the LOREC DSECT and at offset 1A in the LORECX DSECT provides the ID of the nucleus executing the command. In a cluster environment, the nucleus ID is either the SMPID or the PLXID; in a noncluster environment, the nucleus ID is zero (0).

Timestamps in CLOGLAYOUT Records

Timestamps in an Adabas 8 command log created using CLOGLAYOUT=8 are stored in machine time (GMT), whereas CLOGLAYOUT=5 timestamps are stored, as always, in local time. The LORECX record layout that describes the CLOGLAYOUT=8 command log includes a differential time field that stores the difference between machine time and local time at the time the CLOG record is written. This field allows you to calculate the local time of a command log record.

Because of the different timestamp formats used in CLOGLAYOUT=5 and CLOGLAYOUT=8 command logs, we do not recommend that you mix or merge command logs created using different CLOGLAYOUT settings (and, in fact, Adabas does not allow this in cluster or parallel services environments). If you do, your results will be unpredictable.

The command logs used for all Adabas nuclei in a cluster environment must be created using the same CLOGLAYOUT setting. If you mix settings, your results will be unpredictable.

PRILOG : Printing the Command Log

Adabas provides the PRILOG print program to read and report the contents of Adabas command logs in either version 5 or version 8 command log layout format.

PRILOG reads a sequential Adabas command log that has been produced directly by the Adabas nucleus (DD/LOG file) or by the ADARES CLCOPY utility when the Adabas nucleus uses dual or multiple comand logging.

PRILOG is supplied in both source and object form.

Note:
Although PRILOG can be customized, Software AG does not support the program when it has been modified by the user.

In source form, three modules (PRILOG, CCSTCK, and PRILOGD) are supplied for the z/OS, VSE, and BS2000 operating systems. These modules replace all PRILOG versions supplied with earlier versions of Adabas.

Two of the PRILOG modules are system-independent components and one is specific to a particular operating system:

PRILOG interprets control statements; generates report lines from CLOG records.

CCSTCK converts internal timestamp information on CLOG records into a more useful form before printing, making it compliant with Year 2000 standards. CCSTCK is provided independently so it can also be used by other programs.

PRILOGD retrieves an input control card image and a CLOG record and prints a line. PRILOGD is system-dependent. It contains a number of parameters that are described in the source.

This section covers the following topics:

- Print Program Input
- Control Statement Parameters
- Command Selection Parameters
- PRILOG Messages
- Installing and Using PRILOG under VSE
- Installing and Using PRILOG under z/OS

Print Program Input

As input, the PRILOG program requires CLOG records and control statements.

Control statements must begin with the program name "PRILOG" in columns 1 through 7; at least one space must follow the program name before parameters are entered.

Parameters can be entered up to column 71. No continuation or parameter splitting is permitted. Additional parameters can be entered on a separate PRILOG statement.

A comment line begins with an asterisk (*). Comments may also be added to the right side of the parameter string as long as the comment is separated from the parameter value by at least one space.

Control Statement Parameters

Three parameters can be entered on the PRILOG control statements:

- CLOGLAYOUT Parameter
- FIELDS Parameter
- DIMENSIONS Parameter

CLOGLAYOUT Parameter

The CLOGLAYOUT parameter identifies the format of the CLOG records being used as input to the PRILOG program. Valid values are 5 and 8; the default value is 5. The syntax is:

```
CLOGLAYOUT={5 | 8}
```

FIELDS Parameter

The FIELDS parameter identifies the item (or items) from the CLOG records that are to be printed. Its syntax is:

```
FIELDS= { ( item, ... ) | ( LIST) }
```

Replace *item* with one of the following:

Item	Description
ACBX	Adabas extended control block
FB	format buffer
IB	ISN buffer
IOL	I/O list
LIST	Adabas control block field list
MB	multifetch buffer
PB	performance buffer
RB	record buffer
SB	search buffer
UXB	user exit B buffer
VB	value buffer
VERB	verbosity (see below)

The default value is "LIST". Multiple *items* can be listed in any order.

If a data item listed in the FIELDS parameter is not being captured during the ADALOG session and is therefore not present in the CLOG record, the request to print that data item is ignored.

If "VERB" is specified, then a more verbose output is generated. The additional fields displayed are the Additions 1, Additions 2 and Additions 5 fields along with the full communications ID. These values are all displayed in hexadecimal format. Also, if the DIMENSIONS parameter specifies a narrower output format, the fields not printed in the columns will be displayed in the verbose output format.

DIMENSIONS Parameter

The optional DIMENSIONS parameter is used to specify the format that the printed output should take. The number of columns must be 80, 133, or 163, but any number of rows can be specified. If no value for rows is specified, there will be no page breaks and the column headers will print only once. The syntax of the DIMENSIONS parameter:

```
DIMENSIONS=({80[, x] | 133[, x]  | 163[, x] | OLD})
```

If the DIMENSIONS parameter is not specified when CLOGLAYOUT=5 is specified, the traditional 118-column/58-row output is generated. If it is specified, a new style output will be generated according to the dimensions specified.

If the DIMENSIONS parameter is not specified when CLOGLAYOUT=8 is specified, the output will default to 163 columns with no page breaks. If you prefer to generate the traditional 118-column/58-row output format for CLOGLAYOUT=8 logs, specify DIMENSIONS=OLD.

Command Selection Parameters

You can also filter the PRILOG output using the command selection parameters described in this section:

- CLASS Command Selection Parameter
- COMMAND Command Selection Parameter
- DATE Command Selection Parameter
- FILE Command Selection Parameter
- LOGIN_ID Command Selection Parameter
- RECORDS Command Selection Parameter
- RESPONSE Command Selection Parameter
- USER_ID Command Selection Parameter

CLASS Command Selection Parameter

Use the CLASS parameter to select log records whose command codes belong to the specified command classes. The CLASS parameter has the following syntax:

```
CLASS=(keyword[,keyword]...)
```

The following table describes the *keywords* that can be specified:

Keyword	Commands Included
CONTROL	BT, CL, C1, C3, C5, ET, HI, OP, MC, RC, RE, or RI
FIND	S1, S2, S4, S8, or S9
READ	LF, L1, L2, L3, L4, L5, L6, or L9
UPDATE	A1, E1, I1, N1, or N2

In the following example, the commands included in the FIND and UPDATE keywords are selected:

```
PRILOG CLASS=(FIND,UPDATE)
```

COMMAND Command Selection Parameter

Use the COMMAND parameter to select log records whose commands match specified Adabas command names. The COMMAND parameter has the following syntax:

```
COMMAND=(cmd[,cmd]...)
```

Any valid Adabas command can be specified for *cmd*. In the following example, log records for the OP, S1, L1, E1, and CL commands are selected:

```
PRILOG COMMAND=(OP, S1, L1, E1, CL)
```

DATE Command Selection Parameter

Use the DATE parameter to select log records whose timestamps fall between the inclusive start and end times specified. The DATE parameter has the following syntax:

```
DATE=(start-dd-mmm-yyyy[:start-hh:mm:ss][,end-dd-mmm-yyyy[:end-hh:mm:ss]])
```

All times must be specified in military (24-hour) time format.

The first date and time specified in the DATE parameter is the start date and time; the second date and time pair in the syntax is the end date and time. A comma separates these start and end timestamps. Log records with dates that fall on or between these timestamps will be selected. In the following example, all log data occurring between (and including) 19:16:00.000 and 20:20:20.999 on October 10, 2006 will be selected.

```
PRILOG DATE=(10-OCT-2006:19:16:00,10-OCT-2006:20:20:20)
```

If a time is specified, a date is mandatory. If a time is not specified or if only part of the time is specified, the missing values will be assumed to be zero. For example, a specification of 10-OCT-2006:19:16 is interpreted as 10-OCT-2006:19:16:00 and a specification of 10-OCT-2006:19 is interpreted as 10-OCT-2006:19:00:00.

If no start timestamp is specified, then all records that fall up until the end of the specified range are selected. In the following example, all log records are selected that occurred earlier than and at 20:20:20:999 of October 10, 2006:

```
PRILOG DATE=(,10-OCT-2006:20:20:20)
```

If no end timestamp is specified, then all records that fall after the start of the specified range are selected. In the following example, all log records are selected that occur after or at 20:00:00:000 on October 10, 2006.

```
PRILOG DATE=(10-OCT-2006:20:00:00,)
```

If only one date (with no comma) is specified, then the records that are selected must have a timestamp that match the specified timestamp. In the following example, all log records are selected that occur within (and including) the range 10-OCT-2006:19:16:00.000 through 10-OCT-2006:19:16:00.999:

```
PRILOG DATE=(10-OCT-2006:19:16)
```

FILE Command Selection Parameter

Use the file parameter to select log records which reference a specified list of file numbers or range of file numbers. The FILE parameter has the following syntax:

```
FILE=(num[-num][,num[-num]]...)
```

Any valid Adabas file numbers can be substituted for *num*. A maximum of ten ranges or file number values may be specified

In the following example, any log record referencing file numbers 10, 20 or 30 through 40 are selected.

```
PRILOG FILE=(10,20,30-40)
```

LOGIN_ID Command Selection Parameter

Use the LOGIN_ID parameter to select log records with references to a specific login ID. The LOGIN_ID parameter has the following syntax:

```
LOGIN_ID=string
```

Only one login ID can be specified for *string*. In the following example, any log record containing XSCPCOC in the LOGIN_ID column is selected.

```
PRILOG LOGIN_ID=XSCPCOC
```

RECORDS Command Selection Parameter

Use the RECORDS parameter to select log records with references to the specified command sequence number or range of command sequence numbers. The RECORDS parameter has the following syntax:

```
RECORDS=num[-num]
```

Any valid command sequence number can be substituted for *num*. In the following example, any log record containing a command sequence number between and including 101 and 677 will be selected.

```
PRILOG RECORDS=101-677
```

RESPONSE Command Selection Parameter

Use the RESPONSE parameter to select log records that reference the specified list of response codes or range of response codes. THe RESPONSE parameter has the following syntax:

```
RESPONSE=(num[-num][,num[-num]]...)
```

Any valid Adabas response codes can be substituted for *num*. A maximum of ten ranges or response code values may be specified

In the following example, any log record referencing response codes 0, 17, or 15 through 61 are selected.

```
PRILOG RESPONSE=(0,17,15-61)
```

USER_ID Command Selection Parameter

Use the USER_ID parameter to select log records with references to a specific user ID. The USER_ID parameter has the following syntax:

```
USER_ID=string
```

Only one user ID can be specified for *string*. In the following example, any log record containing XSCPCOC in the USER_ID column is selected.

```
PRILOG USER_ID=XSCPCOC
```

PRILOG Messages

PRILOG messages are documented in the *Adabas Messages and Codes* manual as "PL6*nnna*" (independent) and "PL6*ann*" (system-dependent) messages.

Installing and Using PRILOG under VSE

The following components comprise the PRILOG print program for VSE:

Member	Description
PRILOG.A	Independent PRILOG assembly language source module
PRILOGD.A	VSE-dependent PRILOG assembly language source module
CCSTCK.A	Independent Adabas STCK conversion assembly language source module
PRILOG.OBJ	Object deck for PRILOG
PRILOGD.OBJ	Object deck for PRILOGD
CCSTCK.OBJ	Object deck for CCSTCK
PRILOG.PHASE	Executable phase for PRILOG
ASMPRILO.X	Sample VSE JCS to assemble, catalog, and link the PRILOG, CCSTCK, and PRILOGD components into the PRILOG.PHASE
JPRILOG.X	Sample VSE JCS to execute the PRILOG utility

The PRILOG.PHASE may be executed from the library without additional preparation. However, if it becomes necessary to reassemble and relink the PRILOG.PHASE, the following points must be observed:

- The PRILOG.A, CCSTCK.A, and PRILOGD.A modules must be assembled using the IBM high-level Assembler (ASMA90).

- The provided sample JCS member ASMPRILO.X must be modified to suit your site’s requirements. Check the volume, and extent information as well as the library and sublibrary information provided in the JCS members.

Executing PRILOG under VSE

Execute the PRILOG print program by running the PRILOG.PHASE in either a static or dynamic partition.

The PRILOG program uses three files:

File	Description
SYSIPT	PRILOG control card data; may be read from any valid device that can be assigned to SYSIPT
SYSLST	PRILOG report file; may be written to any device that can be assigned to SYSLST
SYS001 (DDCLOG)	Sequential command log input file; this file may reside on any IBM-supported disk device, or it may be read from tape

Notes:

1. The PRILOG control cards must be available from SYSIPT. If on disk or tape, they must be 80-byte records with a fixed record format.
2. The PRILOG report file must be written to SYSLST. If assigned to disk or tape, the output records are 121 bytes in length with a fixed record format. ASA control characters are used for printer control and are in the first byte of each print record.

3. The sequential command log file must be assigned to SYS001. The file may be on disk or tape and may not have a block size greater than 32,760 bytes. The record format is VARBLK.

The provided JPRILOG.X JCS member may be modified according to your installation’s requirements to execute the PRILOG print program. Modify the extent, DLBL, sublibrary, and volume information before submitting the job.

Installing and Using PRILOG under z/OS

The following components comprise the PRILOG print program for z/OS:

Member	**Library**	**Description**
PRILOG	ADA*vrs*.SRCE	Independent PRILOG assembly language source module
PRILOGD	ADA*vrs*.SRCE	z/OS-dependent PRILOG assembly language source module
CCSTCK	ADA*vrs*.SRCE	Independent Adabas STCK conversion assembly language source module
ASMPRILO	ADA*vrs*.JOBS	Sample JCL to assemble and link the PRILOG, CCSTCK, and PRILOGD components into the PRILOG load module.
JPRILOG	ADA*vrs*.JOBS	Sample JCL to execute the PRILOG utility

The PRILOG program uses three files:

Files	**Description**
DDCARD	used for input parameter data; may be any sequential 80-byte record file supported by QSAM.
DDPRINT	used for the output command log report; may be assigned to SYSOUT or to any 121-byte record data set with record format of FBA.
DDCLOGIN	used for the input sequential command log file; must be a sequential file produced by the ADARES CLCOPY utility, or the direct DDLOG sequential file produced by an Adabas nucleus when single command logging is used.

PRILOGC: Printing the Command Log

Adabas provides the PRILOGC print program to read and report the contents of Adabas command logs in either version 5 or version 8 command log layout format.

PRILOGC reads a sequential Adabas command log that has been produced directly by the Adabas nucleus (DD/LOG file) or by the ADARES CLCOPY utility when the Adabas nucleus uses dual or multiple comand logging.

PRILOGC is supplied only in source form.

Note:
Although PRILOGC can be customized, Software AG does not support the program when it has been modified. To use PRILOGC you must at least compile the source provided by Software AG, although no modifications are necessary.

PRILOGC sources are supplied for the z/OS, VSE, z/VM, BS2000, Sun Solaris, and Windows operating systems. These modules replace all PRILOGC versions supplied with earlier versions of Adabas.

Two of the PRILOGC modules are system-independent components, and one is specific to a particular operating system.

Note:
In z/VM environments, solution ADA813L002 must be applied before you can use PRILOGC. For more information about this solution and its documentation, please contact your Software AG support representative.

This section covers the following topics:

- z/OS Library Members
- z/VSE Library Members
- z/VM Library Members
- BS2000 Library Members
- Print Program Input
- Command Selection Parameters
- Installing and Using PRILOGC under z/OS
- Installing and Using PRILOGC under z/VSE
- Installing and Using PRILOGC under z/VM
- Installing and Using PRILOGC under BS2000

z/OS Library Members

For z/OS environments, the following sources are provided:

PRILOGC	Main program logic - written in C.
CCSTCK	Code to convert a STCK value into YYMMDD - written in assembler.
ADABASX/ADACLGX/ADAENV/IODESAM	C header files used by PRILOGC.

The following JCL members are provided in z/OS environments:

ASMPRILC	Sample JCL to assemble the CCSTCK ADABAS STCK convert routine and link-edit the CCSTCK module. Also use this member to compile and link PRILOGC using the IBM/C compiler.
JPRILOGC	Sample JCL to run PRILOGC

z/VSE Library Members

For z/VSE environments, the following sources are provided:

PRILOGC.C	Source for main program logic. Written in C.
CCSTCK.A	Source to convert a STCK value into YYMMDD. Written in assembler.
TIMEPRT.A	Source to convert an 8-byte field containing ’clock ticks’ into a string of the form XXX.YYY MS. This is required on VSE due to the absence of long long support in VSE C. This is used in PRILOGC when populating DURation and SELection fields in the output. Written in assembler.
ADABASX.H ADACLGX.H ADAENV.H IODESAM.H	C header files used by PRILOGC

The following JCL members are provided in z/VSE environments:

ASMPRILC.X	Sample JCS to optionally assemble CCSTCK and TIMEPRT, and compile and link PRILOGC.
JPRILOGC.X	Sample JCS to run PRILOGC

z/VM Library Members

For z/VM environments, the following sources are provided:

PRILOGC C	Source for main program logic. Written in C.
CCSTCK ASSEMBLE	Source to convert a STCK value into YYMMDD. Written in Assembler.
ADABASX H ADACLGX H ADAENV H IODESAM H	C header files used by PRILOGC

The following EXECs are provided in z/VM environments:

PRILOGCC	Sample EXEC to compile PRILOGC.
PRILOGCR	Sample EXEC to run PRILOGC

BS2000 Library Members

For BS2000 environments, the following sources are provided:

PRILOGC	Main program logic - written in C.
CCSTCK	Code to convert a STCK value into YYMMDD - written in assembler.
ADABASX.H/ADACLGX.H/ADAENV.H/IODESAM.H	C header files used by PRILOGC.

The following JCL members are provided in BS2000 environments:

ASMBS2	Sample JCL that can be adapted to assemble the CCSTCK ADABAS STCK convert routine.
C	Compile and link PRILOGC using the IBM/C compiler.
JPRILOGC	Sample JCL to run PRILOGC

Print Program Input

As input, the PRILOGC program requires CLOG records and control statements.

Control statements are read from stdin and take one or more of the following parameters:

```
a: print Additions fields
l: large line size
b: dump with verbosity
t: dump with traditional format - dump records in format used by earlier prilogc with Version 5 CLOGS
c: dump Control Block
f: dump Format Buffer
r: dump Record Buffer
s: dump Search Buffer
u: dump User   Buffer
v: dump Value  Buffer
n: dump ISN    Buffer
6: dump in Open System V6 Format
8: dump in Mainframe V8 Format
x: dump exted  Buffer (User Data)
i: dump IO List
h: dump help information
d: output dimensions (columns,rows), where colums can be 80 or 133 or 163
```

Example: -clfrsvbtnix -d=133,200

If a data item specified from the list above is not being captured during the ADALOG session and is therefore not present in the CLOG record, the request to print that data item is ignored.

If *-b verbosity* is specified, then a more verbose output is generated. The additional fields displayed are the Additions 1, Additions 2, and Additions 5 fields along with the full communications ID. These values are all displayed in hexadecimal format. Also, if the -d DIMENSIONS parameter specifies a narrower output format, the fields not printed in the columns will be displayed in the verbose output format.

The optional DIMENSIONS parameter (-d=cols,rows) is used to specify the format that the printed output should take. The number of columns must be 80, 133, or 163, but any number of rows can be specified. If no value for rows is specified, there will be no page breaks and the column headers will print only once.

Command Selection Parameters

You can also filter the PRILOGC output using the command selection parameters described in this section:

- CLASS Command Selection Parameter
- COMMAND Command Selection Parameter
- DATE Command Selection Parameter
- FILE Command Selection Parameter
- LOGIN_ID Command Selection Parameter
- RECORDS Command Selection Parameter
- RESPONSE Command Selection Parameter
- USER_ID Command Selection Parameter

CLASS Command Selection Parameter

Use the CLASS parameter to select log records whose command codes belong to the specified command classes. The CLASS parameter has the following syntax:

```
CLASS=(keyword[,keyword]...)
```

The following table describes the *keywords* that can be specified:

Keyword	Commands Included
CONTROL	BT, CL, C1, C3, C5, ET, HI, OP, MC, RC, RE, or RI
FIND	S1, S2, S4, S8, or S9
READ	LF, L1, L2, L3, L4, L5, L6, or L9
UPDATE	A1, E1, I1, N1, or N2

In the following example, the commands included in the FIND and UPDATE keywords are selected:

```
CLASS=(FIND,UPDATE)
```

COMMAND Command Selection Parameter

Use the COMMAND parameter to select log records whose commands match specified Adabas command names. The COMMAND parameter has the following syntax:

```
COMMAND=(cmd[,cmd]...)
```

Any valid Adabas command can be specified for *cmd*. In the following example, log records for the OP, S1, L1, E1, and CL commands are selected:

```
PRILOGC COMMAND=(OP, S1, L1, E1, CL)
```

DATE Command Selection Parameter

Use the DATE parameter to select log records whose timestamps fall between the inclusive start and end times specified. The DATE parameter has the following syntax:

```
DATE=(start-dd-mmm-yyyy[:start-hh:mm:ss][,end-dd-mmm-yyyy[:end-hh:mm:ss]])
```

All times must be specified in military (24-hour) time format. Note that the date fields above can optionally be followed by hours, minutes, and seconds.

The first date and time specified in the date parameter is the start date and time; the second date and time pair in the syntax is the end date and time. A comma separates these start and end timestamps. Log records with dates that fall on or between these timestamps will be selected. In the following example, all log data occurring between (and including) 19:16:00.000 and 20:20:20.999 on October 10, 2006 will be selected.

```
DATE=(10-OCT-2006:19:16:00,10-OCT-2006:20:20:20)
```

If a time is specified, a date is mandatory. If a time is not specified or if only part of the time is specified, the missing values will be assumed to be zero. For example, a specification of 10-OCT-2006:19:16 is interpreted as 10-OCT-2006:19:16:00 and a specification of 10-OCT-2006:19 is interpreted as 10-OCT-2006:19:00:00.

If no start timestamp is specified, then all records that fall up until the end of the specified range are selected. In the following example, all log records are selected that occurred earlier than and at 20:20:20:999 of October 10, 2006:

```
DATE=(,10-OCT-2006:20:20:20)
```

If no end timestamp is specified, then all records that fall after the start of the specified range are selected. In the following example, all log records are selected that occur after or at 20:00:00:000 on October 10, 2006.

```
DATE=(10-OCT-2006:20:00:00,)
```

If only one date (with no comma) is specified, then the records that are selected must have a timestamp that match the specified timestamp. In the following example, all log records are selected that occur within (and including) the range 10-OCT-2006:19:16:00.000 through 10-OCT-2006:19:16:00.999:

```
DATE=(10-OCT-2006:19:16)
```

FILE Command Selection Parameter

Use the file parameter to select log records which reference a specified list of file numbers or range of file numbers. The FILE parameter has the following syntax:

```
FILE=(num[-num][,num[-num]]...)
```

Any valid Adabas file numbers can be substituted for *num*. A maximum of ten ranges or file number values may be specified

In the following example, any log record referencing file numbers 10, 20 or 30 through 40 are selected.

```
FILE=(10,20,30-40)
```

LOGIN_ID Command Selection Parameter

Use the LOGIN_ID parameter to select log records with references to a specific login ID. The LOGIN_ID parameter has the following syntax:

```
LOGIN_ID=string
```

Only one login ID can be specified for *string*. In the following example, any log record containing XSCPCOC in the LOGIN_ID column is selected.

```
LOGIN_ID=XSCPCOC
```

RECORDS Command Selection Parameter

Use the RECORDS parameter to select log records with references to the specified command sequence number or range of command sequence numbers. The RECORDS parameter has the following syntax:

```
RECORDS=num[-num]
```

Any valid command sequence number can be substituted for *num*. In the following example, any log record containing a command sequence number between and including 101 and 677 will be selected.

```
RECORDS=101-677
```

RESPONSE Command Selection Parameter

Use the RESPONSE parameter to select log records that reference the specified list of response codes or range of response codes. The RESPONSE parameter has the following syntax:

```
RESPONSE=(num[-num][,num[-num]]...)
```

Any valid Adabas response codes can be substituted for *num*. A maximum of ten ranges or response code values may be specified

In the following example, any log record referencing response codes 0, 17, or 15 through 61 are selected.

```
RESPONSE=(0,17,15-61)
```

USER_ID Command Selection Parameter

Use the USER_ID parameter to select log records with references to a specific user ID. The USER_ID parameter has the following syntax:

```
USER_ID=string
```

Only one user ID can be specified for *string*. In the following example, any log record containing XSCPCOC in the USER_ID column is selected.

```
USER_ID=XSCPCOC
```

Installing and Using PRILOGC under z/OS

The following components comprise the PRILOGC print program for z/OS:

Member	Library	Description
PRILOGC	ADA*vrs*.SRCE	Independent PRILOGC source module written in C.
CCSTCK	ADA*vrs*.SRCE	Independent Adabas STCK conversion assembly language source module.
ASMPRILC	ADA*vrs*.JOBS	Sample JCL to assemble and link CCSTCK, and compile and link PRILOGC load module.
JPRILOGC	ADA*vrs*.JOBS	Sample JCL to execute the PRILOGC utility.

Before PRILOGC can run on z/OS, the PRILOGC load module must be built by running the compile and link step of the sample job ASMPRILC.

The PRILOGC program uses three files:

Files	Description
PARM data	PRILOGC control card parameter data, passed via the PARM= keyword on the JCL EXEC statement.
SYSPRINT	used for the output command log report; may be assigned to SYSOUT or to any 121-byte record data set with record format of FBA.
PRLIN	used for the input sequential command log file; must be a sequential file produced by the ADARES CLCOPY utility, or the direct DDLOG sequential file produced by an Adabas nucleus when single command logging is used.

Installing and Using PRILOGC under z/VSE

The following components comprise the PRILOGC print program for z/VSE:

Member	Description
PRILOGC.C	Independent PRILOGC assembly language source module
CCSTCK.A	Independent STCK conversion assembly language source module
TIMEPRT.A	VSE-only time conversion assembly language source module
PRILOGC.OBJ	Object deck for PRILOGC
CCSTCK.OBJ	Object deck for CCSTCK
TIMEPRT.OBJ	Object deck for TIMEPRT
ASMPRILC.X	Sample VSE JCS to optionally assemble and catalog CCSTCK and TIMEPRT, and compile and link PRILOGC into the PRILOGC.PHASE
JPRILOGC.X	Sample VSE JCS to execute the PRILOGC utility

Before PRILOGC can be run on z/VSE, the PRILOGC.PHASE must be built, using linkage editor control statements from the link-edit step in sample job ASMPRILC.X.

When running ASMPRILC to reassemble PRILOGC components or link the PRILOGC.PHASE, the following points must be observed:

- The CCSTCK.A, and TIMEPRT.A components must be assembled using the IBM high-level Assembler (ASMA90).
- The provided sample JCS member ASMPRILC.X must be modified to suit your site’s requirements. Check the volume and extent information, as well as the library and sublibrary information provided in the JCS members.

Executing PRILOGC under z/VSE

Execute the PRILOGC print program by running the PRILOGC.PHASE in either a static or dynamic partition.

The PRILOGC program uses three files:

File	Description
PARM data	PRILOGC control card parameter data, passed via the PARM= keyword on the JCS EXEC statement.
SYSLST	PRILOGC report file; may be written to any device that can be assigned to SYSLST
SYS001 (PRLIN)	Sequential command log input file; this file may reside on any IBM-supported disk device, or it may be read from tape

Notes:

1. The PRILOGC report file must be written to SYSLST. ASA control characters are used for printer control and are in the first byte of each print record.
2. The sequential command log file must be assigned to SYS001. The file may be on disk or tape and may not have a block size greater than 32,760 bytes. The record format is VARBLK.

The provided JPRILOGC.X JCS member may be modified according to your installation’s requirements to execute the PRILOGC print program. Modify the extent, DLBL, sublibrary, and volume information before submitting the job.

Installing and Using PRILOGC under z/VM

The following components comprise the PRILOGC print program for z/VM:

Member	Description
PRILOGC C	Independent PRILOGC assembly language source module
CCSTCK ASSEMBLE	Independent STCK conversion assembly language source module
PRILOGCC EXEC	Sample EXEC to compile PRILOGC
PRILOGCR EXEC	Sample EXEC to run the PRILOGC utility

To install and run PRILOGC in z/VM environments:

Be sure that solution ADA813L002 is applied before you use PRILOGC. For more information about this solution and its documentation, please contact your Software AG support representative.

1. Run the PRILOGCC EXEC to compile and build PRILOGC.

When running PRILOGCC to recompile PRILOGC components , the following points must be observed:

- The provided sample PRILOGCC EXEC must be modified to suit your site’s requirements.

2. Modify PRILOGCR. Set CLOGFN, CLOGFT, and CLOGFM to the filename, filetype, and filemode of the input Command Log file. Set OUTFN and OUTFT to the filename and filemode of the output file to which the PRILOGC output will be redirected.

3. Run PRILOGCR, providing input parameters if desired.

Installing and Using PRILOGC under BS2000

The following components comprise the PRILOGC print program under BS2000:

Member	Library	Description
PRILOGC (S)	ADA*vrs*.BS2SRCE	Independent PRILOGC source module written in C.
CCSTCK (S)	ADA*vrs*.BS2SRCE	Independent Adabas STCK conversion assembly language source module.
ASMBS2 (J)	ADA*vrs*.BS2SRCE	General sample JCL that can be adapted to assemble.
C (J)	ADA*vrs*.BS2SRCE	Sample JCL to compile PRILOGC and bind it together with CCSTCK and C runtime components.
JPRILOGC (J)	ADA*vrs*.BS2SRCE	Sample JCL to execute the PRILOGC utility.

Before PRILOGC can run in BS2000 environments, the PRILOGC load module must be built by running the compile and link step of the sample job C.

The PRILOGC program reads the control card parameter data from SYSDTA in the job deck and writes the command col report to SYSOUT.

The input sequential command log file produced by the ADARES CLCOPY utility is accessed through the link name PRLIN. To read this, it must be declared as unblocked. For example:

```
/SET-FILE-LINK PRLIN,DB99.CLOG1,REC-FORM=U
```