
**THE INTERPRETIVE SPACE EXECUTIVE
FOR THE
zOS OPERATING SYSTEM**

USER'S GUIDE AND REFERENCE

May 2003

TABLE OF CONTENTS

SUMMARY OF CHANGES	5
PREFACE	6
INTRODUCTION	7
INSTALLATION	8
MACHINE REQUIREMENTS.....	8
SYSTEM REQUIREMENTS	8
VIRTUAL STORAGE REQUIREMENTS.....	8
INSTALLING ISX IN zOS	8
ISX CUSTOMIZATION.....	9
<i>Authorization of the ISX Libraries</i>	9
<i>ISX Startup Job</i>	9
EXECUTION CONTROL	10
COMMUNICATING WITH ISX	10
DEFINITION OF THE GUEST ENVIRONMENT	10
DEFINITION OF ISX LICENSE PARAMETERS.....	11
CONTROLLING REAL I/O DEVICES	11
CONTROLLING EMULATED UNIT RECORD I/O DEVICES	12
CONTROLLING EMULATED FBA DASD I/O DEVICES	13
CONTROLLING EMULATED CKD DASD I/O DEVICES.....	14
CONTROLLING EMULATED MAGNETIC TAPE DEVICES	16
CONTROLLING EMULATED 3270 DEVICES	18
CONTROLLING EMULATED CHANNEL-TO-CHANNEL ADAPTERS.....	22
CONTROLLING EMULATED TOKEN-RING ADAPTERS.....	23
CONTROLLING EMULATED TOKEN-RING ADAPTERS FOR zOS	24
CONTROLLING GUEST EXECUTION.....	26
OPERATOR COMMAND DESCRIPTION	28
NOTATIONAL CONVENTIONS.....	28
ACCOUNT COMMAND	29
ATTACH COMMAND	34
BEGIN COMMAND	35
CAC COMMAND	36
CLOSE COMMAND.....	36
DEFINE COMMAND	37
DETACH COMMAND	42
DISPLAY COMMAND.....	43
DUMP COMMAND	45
EXTERNAL COMMAND	47
HMC COMMAND	47
IPL COMMAND	47
READY COMMAND.....	48
RESET COMMAND	48
RESTART COMMAND.....	49
SET COMMAND	49
SHUTDOWN COMMAND.....	52
STOP COMMAND.....	53
STORE COMMAND.....	53
TAKE COMMAND.....	55
TRACE COMMAND	59
XBEGIN COMMAND	63

XDISPLAY COMMAND.....	63
XDUMP COMMAND.....	64
XSTOP COMMAND.....	65
XSTORE COMMAND.....	66
MESSAGES.....	68

SUMMARY OF CHANGES

This summary of changes includes specific updates to this book as well as product highlights for the Interpretive Space Executive Version 3 Release 1.

The following facilities have been added in the Interpretive Space Executive Version 3 Release 1:

- execution in the zOS operating system,
- simulation of S/370 architecture,
- simulation of S/390 architecture,
- emulation of the hardware management console.

PREFACE

This book describes how to operate the Interpretive Space Executive (ISX) programs. It contains a general explanation of ISX and its capabilities, a description of what you need to install ISX as well as the configuration possibilities, an operator command descriptions, and ISX message information.

INTRODUCTION

THE INTERPRETIVE SPACE EXECUTIVE (ISX) is optional licensed program available for the zOS operating system that offers facilities to run non zOS operating systems and their associated application programs in an zOS environment.

In an zOS address space controlled by ISX, the zOS operating system services are replaced by the operating system services required by the application programs. Several zOS regions can be used by ISX to run different operating systems and their applications simultaneously. Running non-zOS operating systems and applications in zOS address spaces controlled by ISX allows zOS users to work with those applications and operating systems while sharing the zOS system resources with other users.

The naturalization of applications with their own operating system services is greatly simplified due to zOS resource allocation and control facilities and ISX device emulation facilities. The ISX implementation takes full advantage of the zArchitecture Interpretive Execution Mode.

The virtual storage area of an zOS address space used by ISX in the interpretive execution mode of zArchitecture is called an INTERPRETIVE SPACE. An application program together with its native operating system environment, that has to replace zOS in an interpretive space is called a GUEST APPLICATION, or GUEST SYSTEM, or shortly GUEST.

The following figure illustrates how some guest system appears in an interpretive space like a separate job with its own resources assigned by the zOS operating system.

IBM zArchitecture Computer				
zOS				
J E S 2	R A C F	ISX	A P P C	V T A M
		G U E S T		

ISX permits the execution of a guest system, which requires S/370 or S/390 facilities.

INSTALLATION

Before installing ISX, certain requirements and considerations must be reviewed.

Machine Requirements

ISX will run on any IBM zArchitecture processor that has the interpretive execution facility and is supported by the zOS operating system.

One magnetic tape device (e.g. 34xx tape unit) is required to install ISX.

Auxiliary DASD storage space is required for the following:

- SYSPRINT, if routed to DASD,
- 20 tracks of space on a DASD device (such as 3380 or 3390) for the ISX load library.

System Requirements

ISX requires zOS.

Virtual Storage Requirements

The minimum amount of storage required for ISX execution can be estimated using the following requirements:

- 512K is required to run ISX,
- 16K for every I/O device used by a guest system,
- zOS system usage requirements,
- size of main storage as required by a guest system.

Installing ISX in zOS

ISX distribution tape is a standard-labeled magnetic tape written at 6250 bpi with a volume serial number ISXVOL.

To unload the ISX library you must use the IEBCOPY utility. Use the following JCL to unload the ISX library from the install tape. Update the UNIT= and VOL=SER= parameters with values appropriate to your installation:

```
//ISXINST JOB (account),'name',TIME=1440,REGION=1M,
//        CLASS=c,MSGLEVEL=(1,1),MSGCLASS=m,
//        NOTIFY=userid,USER=userid
//ISXSTEP EXEC PGM=IEBCOPY
//SYSUT1 DD DSN=ISX.LINKLIB,DISP=SHR,LABEL=(1,SL),
//        UNIT=TAPE,VOL=SER=ISXVOL
//SYSUT2 DD DSN=ISX.LINKLIB,DISP=(,CATLG),UNIT=uuuu,
//        SPACE=(TRK,(20,2,4)),VOL=SER=xxxxxx,
//        DCB=(RECFM=U,BLKSIZE=32740)
//SYSPRINT DD SYSOUT=*
```

ISX Customization

Authorization of the ISX Libraries

The ISX program library requires Authorized Program Facility (APF) authorization. You will have to add the ISX.LINKLIB library to the appropriate IEAAPFxx or PROGxx PARMLIB member, or you have to use the SETPROG APF,ADD operator command.

The APF list contains the data set names of libraries along with the volume serial numbers of the DASD volumes on which the libraries reside. You must code ISX.LINKLIB name and the volume serial number of the DASD volume where the ISX library resides.

If you are using the IEAAPFxx member, ensure you have the correct APF=xx statement coded in the IEASYSxx parmlib member (or the zOS system operator has the correct APF=xx setting to specify at IPL time).

If you are using the PROGxx member, ensure you have the correct PROG=xx statement coded in the IEASYSxx parmlib member (or the system operator has the correct PROG=xx setting to specify at IPL time). Once the PROGxx member is updated, the library can be dynamically authorized by using the SET PROG=xx operator command.

The ISX library does not have to be included in the linklist.

ISX Startup Job

Use the following JCL to execute an ISX job:

```
//ISXRUN JOB (account),'name',TIME=1440, REGION=nM,
//        CLASS=c,MSGLEVEL=(1,1),MSGCLASS=m,
//        NOTIFY=userid,USER=userid
//ISXSTEP EXEC PGM=pgmname
//SYSLIB DD DSN=ISX.LINKLIB,DISP=SHR
//STEPLIB DD DSN=ISX.LINKLIB,DISP=SHR
//SYSPRINT DD SYSOUT=A
//SYSIN DD DUMMY
```

The 'pgmname' defines the entry point name of ISX. The ISXIN7 name has to be used if S/370 architecture is required for the guest system. The ISXIN9 name has to be used if S/390 architecture is required for the guest system.

The SYSLIB DD statement is required.

The SYSIN DD statement defines the data set that contains the ISX configuration and operational commands. The commands can be entered from an input data stream, a sequential data set or an ISX operator's console.

EXECUTION CONTROL

Communicating with ISX

When ISX is started the initialization procedure performs the following activities:

- sets the ISX address space as non-swappable,
- receives ISX job name, current CPU ID, and ASID of its address space from zOS,
- opens the load module library,
- fixes a few pages in real storage,
- creates communications subtask,
- creates additional subtasks used for I/O device emulation,
- issues the ISX099I message,
- executes commands from the input command file.

An zOS operator can influence the execution ISX by means of ISX commands. Any zOS operator who is working with ISX becomes the ISX operator. As the operator they have control over ISX tuning, resource assignment, and ISX operation.

All ISX messages are sent to the zOS system console. ISX commands are entered from the zOS system console or from the input command file.

Follow these rules when you code ISX commands in the input command file:

- Start each command on a new logical 80-byte record.
- Include at least one blank between each operand.
- You can continue a command on up to 4 lines. ISX assumes a command line is continued if it did find none-blank in column 72.
- ISX ignores columns 73 through 80.

The ISX operator is able to enter ISX commands via the MODIFY command of zOS:

F ISX_job_name,...ISX_command...

Definition of the Guest Environment

The ISX SET command can be entered from the input command file or an zOS console. With the SET command the ISX operator can define:

- whether or not ISX tests each BTAM autopoll CCW to see if it has been dynamically modified.
- whether or not ISX tests each PCI interruption is reflected to the guest system to see if the channel program has been dynamically modified.
- controls whether or not ISX displays the corresponding job name before all messages to the operator.
- controls whether or not ISX issues all messages about operation and I/O exceptions.
- defines the name of the zOS console to be used to communicate with ISX (and emulated printer keyboard of a guest system).

- for ISX controls whether or not ISX writes all inbound/outbound frames of the token-ring and all inbound/outbound messages of the emulated token-ring adapter in the log data set.
- defines the CPU and model identifiers for a guest system.
- defines the initial date for the interpretive space.
- defines the initial clock value for the interpretive space.
- defines the time zone value. The time zone value is added to the zOS current time to define the local time for a guest system.

Before the execution of a guest system the ISX operator must define the size of the interpretive space using the ISX SET command. When the interpretive space has been defined, ISX can begin execution of the guest system.

Definition of ISX License Parameters

To confirm the right to use ISX it is necessary to provide a product password, expiration date, and maximum of guest systems value. Without the proper password, expiration date, and maximum of guest systems value a guest system can not be loaded into an interpretive space for execution. The product password, expiration date, and maximum of guest systems value should have been supplied with the product tape.

The password, expiration date, and maximum of guest systems value must be specified with an ISX SET command. The SET command can be issued from the input command file at ISX start up or from the ISX operator console.

Controlling Real I/O Devices

Usually real I/O devices are required to run a guest system. Real I/O devices are assigned to a guest system with the ISX ATTACH command. If a real I/O device must be used permanently by the guest system, the I/O device must be assigned with an ISX ATTACH command specified in the input command file of the ISX startup job. If a real I/O device is not to be used permanently by the guest system, this device may be assigned with an ISX ATTACH command entered from a console by the ISX operator as needed. Assigned devices may be removed from a guest system with the ISX DETACH command.

A real I/O device can be assigned to a guest system in zOS environment, if:

- an I/O device is in zOS configuration,
- an I/O device is not already assigned to some another zOS address space (for non-DASD devices).

If a real I/O device is not supported by zOS the device has to be defined to zOS as DUMMY. To assign the device to a guest system, its device class must be specified on the ISX ATTACH command.

To assign a 'not operational' I/O device to a guest system, it is necessary to request unconditional attachment on the ISX ATTACH command.

Guest systems that are executing in different ISX address spaces in the zOS environment can work with the same DASD devices simultaneously. If a guest system issues a reserve I/O request, it gains exclusive control of the DASD device until a release I/O request is issued.

12 EXECUTION CONTROL

The RESET command allows the ISX operator to reset the active I/O request of a guest system to a real I/O device.

The READY command allows the ISX operator to examine and set the operational status of the specified real I/O device.

The ACCOUNT command allows the ISX operator to estimate I/O activities of a guest system with real I/O devices.

The DISPLAY command allows the ISX operator to display the dynamic characteristics and activities of real devices.

Controlling Emulated Unit Record I/O Devices

Emulated devices such as unit record devices are usually defined to link guest applications with zOS data structures.

Emulated I/O devices are defined for a guest system with the ISX DEFINE command. If an emulated I/O device must be used permanently by the guest system, this I/O device has to be defined with an ISX DEFINE command specified in the input command file of the ISX startup job. If an emulated I/O device is not to be used permanently by the guest system, this device may be defined with the ISX DEFINE command entered from a console by the ISX operator as needed. Defined devices may be removed from a guest system with the ISX DETACH command.

The DEFINE command allows the ISX operator to define emulated printer keyboard, card punch devices, printers, or card reader devices for a guest system. Hardware management console is emulated by ISX for a guest system.

Only one printer keyboard can be defined. The ISX operator's console will be used to display data routed by a guest system to the printer keyboard. The ISX operator will be able to enter input data for the printer keyboard via the MODIFY command of zOS:

```
F ISX_job_name,>...input data...
```

Or

```
F ISX_job_name,CAC ...input data...
```

The ISX operator will be able to enter input data for the emulated hardware management console via the MODIFY command of zOS:

```
F ISX_job_name,HMC ...input data...
```

For card punch and printer emulation the SYSOUT class has to be specified on the ISX DEFINE command using the CLASS parameter. The corresponding SYSOUT data set is created and allocated to ISX for the device. The SYSOUT data set is unallocated and new a SYSOUT data set is allocated, when the ISX operator issues the CLOSE command.

For card reader emulation input a sequential data set or member of a partitioned data set has to be specified on the ISX DEFINE command. The ISX TAKE command can be used to associate a different zOS data set with the emulated card reader.

The ACCOUNT command allows the ISX operator to estimate I/O activities of a guest system with emulated I/O devices.

The DISPLAY command allows the ISX operator to display the dynamic characteristics and activities of emulated devices.

Controlling Emulated FBA DASD I/O Devices

ISX supports emulated FBA DASD to be used by guest systems as real FBA DASD 9332, 9335, or 9336.

Fixed-length relative record VSAM data sets with a record length of 512 bytes are used for FBA DASD emulation.

If you intend to use emulated FBA DASD for the guest system, you must do the following:

- create a VSAM data set in zOS with the IDCAMS Utility Program,
- prepare the VSAM data set with the ISX ISXFBI Utility Program,
- define the emulated FBA DASD to the guest system with the ISX DEFINE command.

A VSAM data set to be used for FBA DASD emulation in ISX is created with the zOS IDCAMS Utility Program just like any other VSAM data set in zOS. The IDCAMS program may be called via zOS TSO command or from a batch job similar to this:

```
//ISXCR  JOB MSGLEVEL=(1,1),MSGCLASS=H,CLASS=A
//      EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=H
//SYSIN DD *
        DEFINE CLUSTER          -
          (NAME(DISK))          -
          NUMBERED              -
          SUBALLOCATION          -
          RECORDSIZE(512 512)  -
          RECORDS(360036)      -
          VOLUMES(DSKFBA)     -
          SHAREOPTIONS(3)      -
          REUSE)
/*
```

In any case you must specify the VSAM data set name, volume serial number of the volume on which the data set will reside, the space to be allocated for the data set and the data set attributes.

There are no restrictions on the VSAM data set name other than the standard zOS requirements for VSAM data set names.

The amount of space to be allocated for the VSAM data set is defined according to the required FBA DASD size. It is best to specify the space allocation using the RECORDS parameter. The number of records for real FBA DASD is equal to:

- 360036 for 9332 model 400,
- 554800 for 9332 model 600,
- 804714 for 9335 model B01,
- 920115 for 9336 model 10,
- 1672881 for 9336 model 20.

If the number of records specified does not match one of the values specified above, then the value is assumed to be the next lowest value and the associated device will be assumed to be the emulated device. For example, if the specified number of records is less than 554800 and greater than 360036, then the emulated device is assumed to be a device 9332 model 400. If the

14 EXECUTION CONTROL

number of records specified is greater than 1672881 then the emulated device will be a 9336 model 20.

The VSAM data set must reside on one DASD volume and can have only one extent. The SUBALLOCATION parameter must be specified to provide existing of required space on the volume.

The VSAM data set format must be fixed-length relative record, with a record length of 512 bytes. The NUMBERED parameter must be specified to indicate the data set format is relative record. The RECORDSIZE parameter must be specified with the average and maximum record sizes equal to 512.

The REUSE parameter must be specified. The ISX Utility Program ISXFBI requires this function.

The ISX Utility Program ISXFBI is used to prepare the VSAM data set for FBA DASD emulation. This program runs as a batch job in zOS. The JCL for the corresponding job looks like this:

```
//FBA      JOB MSGLEVEL=(1,1),MSGCLASS=H
//          EXEC PGM=ISXFBI
//STEPLIB DD DSN=ISX.LINKLIB,UNIT=SYSDA,DISP=SHR
//SYSPRINT DD SYSOUT=H
//
```

After a VSAM data set is created and prepared for FBA DASD emulation, use the ISX DEFINE command to define the emulated FBA DASD to the guest system. The DEFINE command assigns a device number to the emulated FBA DASD and specifies the name of the VSAM data set to be used for this FBA DASD emulation.

Controlling Emulated CKD DASD I/O Devices

ISX supports emulated CKD DASD to be used by guest systems as real CKD DASD 3380 or 3390. The control unit represents as 3990 Model 2. The logical volumes are emulated 3380 model E, J, or K volume and 3390 model 1, 2, 3, or 9 volumes. (With standard 3380/3390 track and cylinder format with fewer cylinders.)

ISX does not support the following commands and facilities of the 3990 Model 2 storage control unit:

- 3380 track compatibility mode
- Alternate/defective Track Facility
- Device Support Track Facility
- Diagnostic Track Facility
- Diagnostic Control
- Buffered Log Data
- Alternate Pathing:
 - Write Inhibit
 - Fencing/Unfencing

- Service Information Message (SIM)

Fixed-length relative record VSAM data sets with a record length not less than 3104 bytes are used for CKD DASD emulation. The recommended record length is 25542 bytes for emulated 3380 and 30211 bytes for emulated 3390.

If you intend to use emulated CKD DASD for the guest, you must do the following:

- create a VSAM data set in zOS with the IDCAMS Utility Program,
- prepare the VSAM data set with the ISX ISXCKI Utility Program,
- define the emulated CKD DASD to the guest system with the ISX DEFINE command.

A VSAM data set to be used for CKD DASD emulation in ISX is created with the zOS IDCAMS Utility Program just like any other VSAM data set in zOS. The IDCAMS program may be called via zOS TSO command or from a batch job similar to this:

```
//ISXCR  JOB MSGLEVEL=(1,1),MSGCLASS=H,CLASS=A
//      EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=H
//SYSIN DD *
        DEFINE CLUSTER          -
          (NAME(DISK)           -
          NUMBERED              -
          SUBALLOCATION          -
          RECORDSIZE(30211 30211) -
          RECORDS(33390)       -
          VOLUMES(DSKCKD)      -
          SHAREOPTIONS(3)      -
          REUSE)
/*
```

In any case you must specify the VSAM data set name, volume serial number of the volume on which the data set will reside, the space to be allocated for the data set and the data set attributes.

There are no restrictions on the VSAM data set name other than the standard zOS requirements for VSAM data set names.

The amount of space to be allocated for the VSAM data set is defined according to the required CKD DASD size. It is best to specify the space allocation using the RECORDS parameter. The number of records with record size 25542 or 30211 for emulated CKD DASD 3380 or 3390 is equal to:

- 26550 for 3380 model J,
- 53100 for 3380 model E,
- 79650 for 3380 model K,
- 33390 for 3390 model 1,
- 66780 for 3390 model 2,
- 100170 for 3390 model 3,
- 300510 for 3390 model 9.

If the number of records specified does not match one of the values specified above, then the value is assumed to be the next lowest value and the associated device will be assumed to be the emulated device. For example, if for emulated 3390 the specified number of records is less than

66780 and greater than 33390, then the emulated device is assumed to be a device 3390 model 2.

The VSAM data set must reside on one DASD volume and can have only one extent. The SUBALLOCATION parameter must be specified to provide existing of required space on the volume.

The VSAM data set format must be fixed-length relative record with a record length that is not less than 3104 bytes. The NUMBERED parameter must be specified to indicate the data set format is relative record. The RECORDSIZE parameter must be specified with the average and maximum record sizes equal to the record length.

The REUSE parameter must be specified. The ISX Utility Program ISXCKI requires this function.

The ISX Utility Program ISXCKI is used to prepare the VSAM data set for CKD DASD emulation. This program runs as a batch job in zOS. The JCL for the corresponding job looks like this:

```
//CKD      JOB MSGLEVEL=(1,1),MSGCLASS=H
//          EXEC PGM=ISXCKI
//STEPLIB DD DSN=ISX.LINKLIB,UNIT=SYSDA,DISP=SHR
//SYSPRINT DD SYSOUT=H
//
```

Just after the ISX Utility Program ISXCKI has been executed there are no volume labels on the volume. Each track contains a home address and record 0 is formatted on all tracks. There are no alternate tracks on the volume. Use the INIT command of the ICKDSF utility program at the minimal level to write a volume label, a VTOC, and other associated information on the volume.

After a VSAM data set is created and prepared for CKD DASD emulation, use the ISX DEFINE command to define the emulated CKD DASD to the guest system. The DEFINE command assigns a device type and device number to the emulated CKD DASD and specifies the name of the VSAM data set to be used for this CKD DASD emulation.

Controlling Emulated Magnetic Tape Devices

ISX supports emulated magnetic tape to be used by guest systems as real 3480 or 3490 devices.

VSAM data sets with the fixed-length relative records and record length of 4096 bytes are used for magnetic tape emulation.

If you intend to use emulated magnetic tape for the guest, you must do the following:

- create a VSAM data set in zOS with the IDCAMS Utility Program,
- prepare the VSAM data set with the ISX ISXDSI Utility Program,
- define the emulated magnetic tape to the guest system with the ISX DEFINE command.

A VSAM data set to be used for the tape emulation in ISX is created with the zOS IDCAMS Utility Program just like any other VSAM data set in zOS. The IDCAMS program may be called via zOS TSO command or from a batch job similar to this:

```
//ISXCR  JOB MSGLEVEL=(1,1),MSGCLASS=H,CLASS=A
//      EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=H
//SYSIN DD *
      DEFINE CLUSTER -
          (NAME(TAPE)           -
           NUMBERED             -
           SUBALLOCATION         -
           RECORDSIZE(4096 4096) -
           RECORDS(1000)       -
           VOLUMES(DSKTAP)     -
           SHAREOPTIONS(3)     -
           REUSE)
/*
```

In any case you must specify the VSAM data set name, volume serial number of the volume on which the data set will reside, the space to be allocated for the data set and the data set attributes.

There are no restrictions on the VSAM data set name other than the standard zOS requirements for VSAM data set names.

The amount of space to be allocated for the VSAM data set is defined according to the required tape size, but not less than 32. It is best to specify the space allocation using the RECORDS parameter.

The VSAM data set must reside on one DASD volume and can have only one extent. The SUBALLOCATION parameter must be specified to provide existing of required space on the volume.

The VSAM data set format must be fixed-length relative record, with a record length of 4096 bytes. The NUMBERED parameter must be specified to indicate the data set format is relative record. The RECORDSIZE parameter must be specified with the average and maximum record sizes equal to 4096.

The REUSE parameter must be specified. The ISX Utility Program ISXDSI requires this function.

The ISX Utility Program ISXDSI is used to prepare the VSAM data set for magnetic tape emulation. This program runs as a batch job in zOS. The JCL for the corresponding job looks like this:

```
//TAPE  JOB MSGLEVEL=(1,1),MSGCLASS=H
//      EXEC PGM=ISXDSI
//STEPLIB DD DSN=ISX.LINKLIB,UNIT=SYSDA,DISP=SHR
//SYSPRINT DD SYSOUT=H
//
```

After a VSAM data set is created and prepared for tape emulation, use the ISX DEFINE or ISX TAKE commands to define the emulated tape to the guest system. The DEFINE command assigns a device number to the emulated magnetic tape and optionally specifies the name of the VSAM data set to be used for the tape emulation. The TAKE command assigns new VSAM data set to be used for the tape emulation.

Controlling Emulated 3270 Devices

ISX supports emulated 3270 display units to be used by guest systems as real local non-SNA 3270 terminals. The emulated 3270 display units look like a 3274 Model 1D with Model 1, 2, 3, 4 or 5 devices without QUERY capability. ISX provides the capability of using emulated 3270 display units through VTAM in zOS.

To use emulated 3270 display units with VTAM the following must be defined:

- ISX has to be defined to VTAM as the application program within an application program major node using an APPL definition statement.
- ISX VTAM session parameters have to be defined through logon mode tables and through VTAM and NCP definition statements.
- A logical unit name of the VTAM terminal has to be defined for an emulated 3270 display unit.

The ISX application program major node must be defined with the VBUILD definition statement and the ISX application program minor nodes must be defined with the APPL definition statement. Each ISX application program to be used must be defined with an APPL definition statement. If you will be running 3 guest systems then there will be 3 APPL statements to define the ISX jobs.

For example,

ISXNODE	VBUILD	TYPE=APPL	
APPLISXA	APPL	AUTH=ACQ,	permits to acquire LUs
		EAS=3,	concurrent sessions
		ACBNAME=APPLISXA,	application name
		SRBEXIT=YES	authorized to use SRB

Note that ISX, as a VTAM application program does not specify its name in the APPLID operand of the ACB macro. Therefore VTAM will use the job step name of the ISX startup job as the application program name. The name of the APPL definition statement must match the job step name of the ISX startup job.

ISX as the VTAM application program supports emulated 3270 display units using logical units of type 0 or 2. The zOS provides a default logon mode table for LU type 0 and LU type 2 3270 devices. This table can be used to create your own logon mode table. The MODETAB and MODEENT macros can be used to define 3270 characteristics in the logon mode table.

The following are examples of table entry definitions:

The MODEENT macro for Non-SNA 3270.

```
name MODEENT FMPROF=X'02',
           TSPROF=X'02',
           PRIPROT=X'71',
           SECPROT=X'40',
           COMPROT=X'20000',
           PSERVIC=X'.....' see Note 1
```

The MODEENT macro for SNA 3270.

```
name MODEENT FMPROF=X'03',
           TSPROF=X'03',
           PRIPROT=X'B1',
           SECPROT=X'90',
           COMPROT=X'3080',
           RUSIZES=X'....',
           PSERVIC=X'.....' see Note 1
```

Note 1. Coding the Device-Specific Hexadecimal Data of the PSERVIC operand.

Hexadecimal code	Related device
X'00..00000000.....00'	For non-SNA (LU 0)
X'02..00000000.....00'	For SNA (LU 2)
X'..00.....'	Device without extended data stream capability
X'..80.....'	Device with extended data stream capability (it is not supported)
X'.....0000000001..'	Buffer size 480 (12X40) only
X'.....0000000002..'	Buffer size 1920 (24X80) only
X'.....0C280C507F..'	for 480 or 960 (12X40 or 12X80)
X'.....185020507F..'	for 1920 or 2560 (24X80 or 32X80)
X'.....18502B507F..'	for 1920 or 3440 (24X80 or 43X80)
X'.....18501B847F..'	for 1920 or 3564 (24X80 or 27X132)
X'.....185000007E..'	for 1920 (24x80) only

To prevent screen size switching on a device that has more than one possible size, code the screen size as: X'.....185000007E..' in the PSERVIC operand.

Note that the alternate screen size defined by the display model number in the guest system has to correspond to the alternate screen size defined by PSERVIC operand.

The FMPROF, TSPROF, PRIPROT, SECPROT and COMPROT values have to be the same as shown in the examples.

The ISX DEFINE command is used to assign the logical unit name of a VTAM terminal to an emulated 3270 display unit.

Several emulated 3270 display units can be defined with the same logical unit name. In this case the first defined display unit will be operational for the guest operating system and connected with the corresponding VTAM terminal. All other display units with the same logical unit name will be operational, but are set up in disconnect mode with respect to the VTAM terminal. The ISX TAKE command allows a VTAM terminal to switch from one emulated device to another. To switch from one emulated display unit to another it is necessary to define a functional key for VTAM terminal switching. This functional key must be defined for every display unit that needs to use the switching function. Once switching occurs the display unit remains operational, but is set up in disconnect mode with respect to the VTAM terminal.

In addition to the VTAM terminal switch capability the TAKE command permits you to redefine the logical unit name for an emulated display unit while the device remains operational to the guest operating system. This capability of the ISX TAKE command permits the switching of an emulated display unit from one VTAM terminal to another.

If an I/O error or unusual condition (such as VTAM termination) are detected while an emulated display unit is connected with a VTAM terminal, then the guest display unit is automatically set up in disconnect mode, but remains operational. The ISX TAKE command permits the emulated display unit to reconnect with the same or another VTAM terminal.

You can request a connection with an emulated display unit via the VTAM LOGON command specifying the ISX application program name (the job step name of the ISX startup job) in the APPLID operand of the VTAM LOGON command.

The VTAM LOGON command in conjunction with the terminal disconnection facility of the ISX TAKE command allows one VTAM terminal to connect with emulated 3270 display units of different guest operating systems.

The following is the first LOGON panel:

```

-----ISX LOGON-----

Enter LOGON parameters below:

Gdevaddr  ===>  _          PASSWORD  ===>

message area

PF3-EXIT  PF9-CANCEL  ENTER-SAVE

```

Gdevaddr value defines the guest device number to connect the VTAM terminal with. The PASSWORD value has to be entered, if it was specified in the ISX DEFINE command for the Gdevaddr. Passwords will not appear when typed. The 'message area' of the panel is designed to display the following LOGON procedure messages:

- Terminal is assigned to the guest device
- Invalid guest device number
- Invalid password
- Device has predefined assignment

The PF3 key is used to connect the VTAM terminal with the guest emulated display units assigned during the LOGON procedure. The PF9 key is used to cancel previous assignments of the emulated display units and terminate the LOGON procedure. The ENTER key is used to go to the next panel of the LOGON procedure, if the guest device number and password are accepted.

The next panel is displayed with the guest functional key definitions:

```

-----ISX LOGON-----

Enter LOGON parameters below:

Gdevaddr ==> 001          PASSWORD ==>

Key definitions:

PF1  ==> PF1  PF2  ==> PF2  PF3  ==> PF3
PF4  ==> PF4  PF5  ==> PF5  PF6  ==> PF6
PF10 ==> PF10 PF11 ==> PF11 PF12 ==> PF12
PF13 ==> PF13 PF14 ==> PF14 PF15 ==> PF15
PF16 ==> PF16 PF17 ==> PF17 PF18 ==> PF18
PF19 ==> PF19 PF20 ==> PF20 PF21 ==> PF21
PF22 ==> PF22 PF23 ==> PF23 PF24 ==> PF24
PA1  ==> PA1  PA2  ==> PA2  PA3  ==> PA3
ENTER ==> ENTER

message area

PF3-EXIT PF9-CANCEL ENTER-SAVE

```

Usually the functional keys of an emulated display unit correspond to the functional keys of the real terminal. The panel is designed to redefine functional keys or to assign special functions to the functional keys of the real terminal. The current key definitions are shown by the symbol ==> on the panel.

Special functions are defined by the following keywords:

- **SYSREQ** - functional key. The result of pressing the real TESTREQ or SYSREQ key is intercepted by the network and is not received by ISX. To emulate the TESTREQ or SYSREQ key for the guest display unit these keys have to be defined through the another functional key.
- **SWITCH** - terminal switch. In order to use the terminal with different emulated display units it is necessary to have the terminal switch function. This function can be assigned to any real functional key of the real terminal.
- **LEAVE** - terminal disconnection. In order to use the terminal with emulated display units of different guest systems it is necessary to have the terminal disconnection function. This function can be assigned to any functional key of the real terminal.
- **BREAK** - terminal power off. In order to emulate the power off conditions at the device it is necessary to have the terminal power off function. This function can be assigned to any functional key of the real terminal. Terminal power on is provided with the reconnect function of the ISX TAKE command or the LOGON procedure.

The 'message area' of the panel is designed for the following LOGON procedure messages:

- Terminal is assigned to the guest device
- Confirm or enter key definitions
- Invalid key definition

The PF3 key is used to connect the VTAM terminal with the guest emulated display unit. The PF9 key is used to cancel previous assignments of the emulated display units and terminate the LOGON procedure. The ENTER key is used to save parameters and to return to the first panel to define the connection of the terminal with the next guest emulated display unit or to terminate the LOGON procedure. The last emulated display unit is connected with the terminal while all other emulated displays become disconnected.

Controlling Emulated Channel-To-Channel Adapters

ISX supports emulated channel-to-channel adapters. Both sides of the emulated channel-to-channel adapter can be connected with the same ISX job or with different ISX jobs. In the last case the channel-to-channel adapter connects two different guest systems that are running in zOS.

If you intend to use an emulated channel-to-channel adapter in the guest system, you must define it with the ISX DEFINE command. If you need to connect the guest system with itself, then you have to specify guest device numbers for both sides of the adapter and the name of the ISX job used to run the guest system in zOS. When connecting two guest systems via the adapter, the ISX DEFINE command for the owning system must specify the guest device numbers of both sides of the adapter and the name of the ISX job that is running the partner system in zOS. The ISX DEFINE command for the partner system must specify the guest device numbers of both sides of the adapter and the name of the ISX job running the owning guest system in zOS.

Controlling Emulated Token-Ring Adapters

ISX supports emulated token-ring adapters that look like an IBM Token-Ring Subsystem (IEEE 802.5) of the IBM 9221 Processors. The adapters can be connected with a real token-ring via the following methods:

- real IBM non-EMIF LAN Channel Station (LCS) devices of OSA/1 or OSA/2 adapter,
- real IBM Link Services Architecture (LSA) device of OSA/2 adapter,
- a channel attached device such as 3172,
- the token-ring can be fully emulated.

In the last case the adapters are connected with the virtual token-ring that is available for all guest systems that are running in one or more zOS systems.

If you intend to use an emulated token-ring adapter in the guest system, it must be defined with the ISX DEFINE command. If you need to connect the guest system with a real token-ring, then you must specify the real token-ring adapter in the ISX DEFINE command. The real token-ring adapter will be dynamically allocated to ISX in zOS and the emulated token-ring adapter receives its MAC address (this MAC address can not be changed by the guest system). When connecting a guest system with other guest systems via the virtual token-ring, a suitable MAC address must be specified in the ISX DEFINE command. This MAC address will be used as the default for the full emulated token-ring adapter. Suitable LU names must be specified in the TAKE commands for all destination MAC addresses of the virtual token-ring. You can define both types of emulated token-ring adapters for the same guest system.

The data transfer rate of an emulated token-ring adapter essentially depends on the time interval to read inbound and outbound request parameters of the guest system. The polling time interval can be defined from 32 to 1 millisecond in the ISX DEFINE command.

ISX supports emulated token-ring adapters that look like an IBM non-EMIF LAN Channel Station (LCS) devices of OSA/1 or OSA/2 adapter. The adapters can be connected with the virtual token-ring.

ISX provides the capability of using an emulated token-ring adapter through the VTAM in zOS.

To use full emulated token-ring adapters the following definitions have to be done to establish LU 6.2 sessions between ISX applications:

- If full emulated token-ring adapters of ISX are located in different zOS systems, then the corresponding network definitions have to be provided in VTAM.
- Every ISX has to be defined to VTAM as the SNA LU 6.2 application program within an application program major node using an APPL definition statement.
- ISX VTAM session parameters are determined by logon mode name ISXLMODE. The parameters must correspond to session characteristics of the partner system.

The ISX application program major node must be defined with the VBUILD definition statement and ISX application programs minor nodes must be defined with one APPL definition statement for each ISX application program in the major node.

For example,

ISXNODE	VBUILD	TYPE=APPL	
APPLISXA	APPL	AUTH=ACQ, EAS=9,	permits to acquire LUs concurrent sessions (should include LU 6.2 and non LU 6.2 sessions)
		ACBNAME=APPLISXA, SRBEXIT=YES, APPC=YES, ATNLOSS=ALL,	application name authorized to use SRB can issue APPCCMD macro ATTN exit is to be scheduled for all session deactivation
		DLOGMOD=ISXLMODE, DMINWNL=1,	logon mode table entry minimum number of contention winner session for logmode
		DMINWNR=1,	minimum number of contention loser session for logmode
		DSESLIM=2,	maximum number of sessions between the local LU and remote LU for logmode
		MODETAB=ISXTAB	both ISXTAB and ISTINCLM are used

Note that ISX, as a VTAM application program does not specify its name in the APPLID operand of the ACB macro. Therefore VTAM will use the job step name of the ISX startup job as the application program name. The name of the APPL definition statement must match the job step name of the ISX startup job.

ISX opens two SNA LU 6.2 sessions for each destination node. That is why, the maximum number of sessions (DSESLIM) between the local LU and remote LU must be not less than 2 and the minimum number of contention winner and contention loser sessions must be not less than 1.

The logon mode table entry specified by the APPL DLOGMOD parameter must be defined. To create your own logon mode table the MODETAB and MODEENT macros have to be used.

```
ISXTAB    MODETAB
ISXLMODE MODEENT LOGMODE=ISXLMODE,TYPE=0,FMPROF=X'13',
                TSPROF=X'07',PRIPROT=X'B0',SECPROT=X'B0',
                COMPROT=X'D0B1',RUSIZES=X'9797',
                ENCR=B'0000',
                PSERVIC=X'060200000000000000002300'
        MODEEND ,
        END
```

The logon mode table must be assembled and link-edited into the library associated with the VTAMLIB DD statement of VTAM startup procedure. This library must also be APF authorized.

Controlling Emulated Token-Ring Adapters for zOS

ISX supports emulated token-ring adapters that look like the LSA or LCS Interface of a processor attached platform such as OSA or a channel attached platform such as a 3172 to

connect emulated token-ring adapters of the virtual token-ring with the host zOS. For emulation of the LSA Interface one real channel-to-channel adapter is required. For emulation of the LCS Interface two real channel-to-channel adapters are required.

If you intend to use an emulated token-ring adapter for connection to the host zOS, you must:

- Prepare a job to run the LSA Emulator of ISX - ISX/LSA or prepare a job to run the LCS Emulator of ISX – ISX/LCS on the host zOS,
- For LSA adapter assign in host zOS one side of the real channel-to-channel adapter to ISX/LSA. For LCS adapter assign in host zOS one side of both real channel-to-channel adapters to ISX/LCS,
- For LSA adapter assign in host zOS the opposite side of the real channel-to-channel adapter to VTAM as a LSA adapter. For LCS adapter assign in host zOS the opposite sides of real channel-to-channel adapters to TCP/IP as a LCS adapter.

Use the following JCL to start up ISX/LSA:

```
//LSA JOB (account),'name',TIME=1440,
// CLASS=c,MSGLEVEL=(1,1),MSGCLASS=m,
// NOTIFY=userid,USER=userid
//APPLLSA EXEC PGM=ISXLSA
//SYSLIB DD DSNAME=ISX.LINKLIB,DISP=SHR
//STEPLIB DD DSNAME=ISX.LINKLIB,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
```

Use the following JCL to start up ISX/LCS:

```
//LCS JOB (account),'name',TIME=1440,
// CLASS=c,MSGLEVEL=(1,1),MSGCLASS=m,
// NOTIFY=userid,USER=userid
//APPLLSA EXEC PGM=ISXLCA
//SYSLIB DD DSNAME=ISX.LINKLIB,DISP=SHR
//STEPLIB DD DSNAME=ISX.LINKLIB,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
```

The job step name of the ISX/LSA startup job has to be defined in the host VTAM as the name of the SNA LU 6.2 application program within an application program major node using an APPL definition statement. The definition statement has to be the same as for the ISX SNA LU 6.2 application program.

The ISXLSA name is the ISX/LSA initialization load module name. The ISXLCA name is the ISX/LCS initialization load module name.

The SYSLIB DD statement is required.

The SYSIN DD statement defines the data set that contains the ISX/LSA or ISX/LCS command lines. The commands can be entered from the input data stream, from a sequential data set or from the ISX/LSA or ISX/LCS operator's console. In any case, the following commands are required:

- The DEFINE command - to define the MAC address of the emulated adapter and to allocate the chosen side(s) of the channel-to-channel adapter(s) to ISX/LSA or ISX/LCS,
- The TAKE command - to associate outbound data of the emulated adapter that are received from VTAM with an emulated adapter of the virtual token-ring,
- The BEGIN command - to start the LSA Interface emulation process.

Both sides of the real channel-to-channel adapter(s) you intend to use, have to be defined in zOS as CTC devices.

For LSA adapter you have to prepare an XCA major node definition in zOS for the connection of VTAM to the virtual token-ring. The XCA major node has to be defined in the same manner as the 3172 interconnected controller. You have to code in the PORT definition statement ADAPNO=0 and CUADDR=xxxx, where xxxx is the device number of the opposite side of the channel-to-channel adapter. For the peripheral nodes being connected to the virtual token-ring you have to prepare the suitable switched major nodes.

For LCS adapter you have to prepare the appropriate DEVICE and LINK statements to the configuration data set PROFILE.TCPIP. The DEVICE statement is used to specify the name and hexadecimal address of the LCS device. It is the device number of the first channel-to-channel adapter. The second channel-to-channel adapter is also used as LCS adapter. The LINK statement is used to define the token-ring link (IBMTR) to the LCS previously defined by the LCS DEVICE statement. You must also specify a START statement to start the device.

Controlling Guest Execution

A guest system can be loaded with the ISX IPL command. The ISX IPL command may be specified in the input command file of the ISX startup job or be entered from the ISX console.

ISX provides commands to simulate S/370 or S/390 hardware interruptions to the guest system. The ISX operator can use:

- The EXTERNAL command to simulate an external key interruption to the guest system.
- The RESTART command to restart the guest system in the interpretive space.

The SHUTDOWN command allows the operator to terminate all ISX activities.

Several ISX commands may be used to locate, display, and alter program instructions and data of the guest system in the interpretive space.

In the following discussions of the STOP, DISPLAY, STORE and DUMP commands, it is important to understand the levels of storage specified on the command line. The first level storage is the real storage of the computer. The second level storage, i.e. the interpretive space storage, is 'real' storage of a guest system. This includes the V=R partitions of VSE/ESA systems, V=R regions of MVS/XA systems. The V=V partitions of VSE/ESA systems or regions of MVS/XA systems constitute the third level storage.

To stop the execution of the guest system, an empty input or the STOP command without any operands may be used. When ISX stops execution of the guest system,, it is possible to enter other commands to display and alter the storage, registers or PSW. To resume execution of the guest system at the address in the current PSW it is necessary to enter the BEGIN command without any operands. To resume execution of the guest system at a specific address, specify the address in the BEGIN command. To stop execution of the guest system at a specific address, use the STOP command with the address where execution is to stop.

The contents of the interpretive space storage, storage protection keys, registers and PSW can be displayed on the terminal using the DISPLAY command or can be written to a special SYSOUT - data set using the DUMP command.

The contents of the interpretive space storage, guest registers, and the PSW can be altered with the STORE command. The STORE command can be used to store the guest CPU status in corresponding guest storage locations.

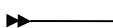
The ACCOUNT command allows the ISX operator to estimate ISX job CPU time, guest CPU time, the number of SIE instructions that were issued by ISX, the number of SSCH or SIO instructions and the number of long channel programs that were issued by a guest system.

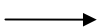
OPERATOR COMMAND DESCRIPTION


The commands described in this document allow the operator to manage the resources of the system, to control guest system execution, to reconfigure ISX parameters and I/O devices assigned to the guest system, and to simulate other functions of S/370 or S/390.


Notational Conventions

Throughout this book, command syntax is described using the structure defined below. Read the syntax diagrams from left to right, from top to bottom, following the path of the line.

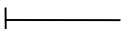
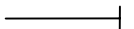
The  symbol indicates the beginning of a statement.

The  symbol indicates that the statement syntax is continued.

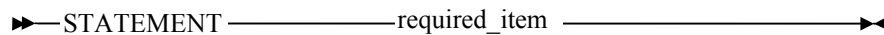
The  symbol indicates that a line is continued from the previous line.

The  symbol indicates the end of a statement.

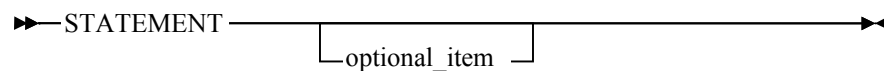
Diagrams of syntactical units other than complete statements

start with the  symbol and end with the  symbol.

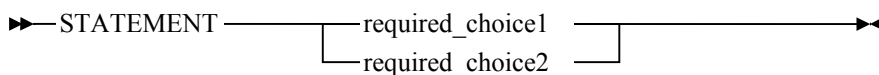
Required items appear on the horizontal line (the main path).



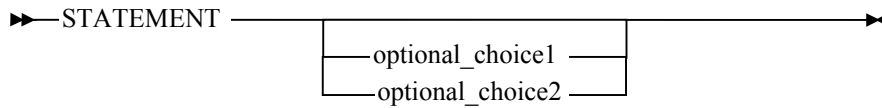
Optional items appear below the main path.



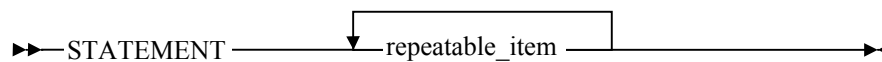
When you can choose from two or more items, they are stacked vertically. If you must choose one of the items, an item of the stack appears on the main path.



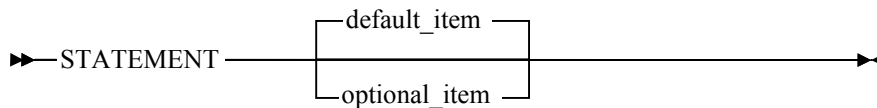
If choosing one of the items is optional, the entire stack appears below the main path.



An arrow returning to the left above the main line indicates an item that can be repeated.



Defaults are above the line. The system uses the default unless you override it. You can override the default by coding an option from the stack below the line.



Uppercase letters of an item denote the shortest acceptable abbreviation. Variables appear in italic lowercase letters (for example, *gdevaddr*). They represent user-supplied names or values. If punctuation marks, parentheses, arithmetic operators, or such symbols are shown, they must be entered as part of the syntax.

ACCOUNT Command

The ACCOUNT command is designed to display ISX or ISX/LSA or ISX/LCS account data on the operator console.

The ACCOUNT command issued for ISX gives four data lines.

The first data line displays the following account data:

- total time the account function was operative (R-TIME),
- total ISX job CPU time the account function was operative (T-TIME),
- total guest CPU time the account function was operative (G-TIME),
- total number of SIE instructions the account function was operative (SIE),
- total number of SSCH/SIO instructions for real devices the account function was operative (R-SIO),
- total number of SSCH/SIO instructions for emulated devices the account function was operative (E-SIO),
- total number of long channel programs for real devices the account function was operative (LCP).

30 OPERATOR COMMAND DESCRIPTION

The second data line displays the following account data:

- total ISX job CPU time utilization the account function was operative (T-TIME),
- total guest CPU time utilization the account function was operative (G-TIME),
- total SIE instruction rate the account function was operative (SIE),
- total SSCH/SIO instruction rate for real devices the account function was operative (R-SIO),
- total SSCH/SIO instruction rate for emulated devices the account function was operative (E-SIO),
- total long channel program rate for real devices the account function was operative (LCP).

The third data line displays the following account data:

- time the previous ACCOUNT command was issued (R-TIME),
- ISX job CPU time from the time the previous ACCOUNT command was issued (T-TIME),
- guest CPU time from the time the previous ACCOUNT command was issued (G-TIME),
- the number of SIE instructions from the time the previous ACCOUNT command was issued (SIE),
- the number of SSCH/SIO instruction for real devices from the time the previous ACCOUNT command was issued (R-SIO),
- the number of SSCH/SIO instructions for emulated devices from the time the previous ACCOUNT command was issued (E-SIO),
- the number of long channel programs for real devices from the time the previous ACCOUNT command was issued (LCP).

The fourth data line displays the following account data:

- ISX job CPU time utilization from the time the previous ACCOUNT command was issued (T-TIME),
- guest CPU time utilization from the time the previous ACCOUNT command was issued (G-TIME),
- SIE instruction rate from the time the previous ACCOUNT command was issued (SIE),
- SSCH/SIO instruction rate for real devices from the time the previous ACCOUNT command was issued (R-SIO),
- SSCH/SIO instruction rate for emulated devices from the time the previous ACCOUNT command was issued (E-SIO),
- long channel program rate for real devices from the time the previous ACCOUNT command was issued (LCP).

The ACCOUNT command issued for ISX/LSA displays four data lines.

The first data line displays the following account data:

- total ISX/LSA job CPU time the account function was operative (T-TIME),
- total number of LU 6.2 RECEIVE operations the account function was operative (RECEIVE),

- total number of inbound frames received from the virtual token-ring the account function was operative (I-FRAME),
- total number of LU 6.2 SEND operations the account function was operative (SEND),
- total number of outbound frames sent to the virtual token-ring the account function was operative (O-FRAME),
- total number of SSCH/SIO instructions for real CTCA the account function was operative (SIO),
- total number of retry I/O operations for real CTCA the account function was operative (RETRY).

The second data line displays the following account data:

- total ISX/LSA job CPU time utilization the account function was operative (T-TIME),
- total LU 6.2 RECEIVE operations frequency the account function was operative (RECEIVE),
- total inbound frames frequency the account function was operative (I-FRAME),
- total LU 6.2 SEND operations frequency the account function was operative (SEND),
- total outbound frames frequency the account function was operative (O-FRAME),
- total SSCH/SIO instructions frequency for real CTCA the account function was operative (SIO),
- total retry I/O operations frequency for real CTCA the account function was operative (RETRY).

The third data line displays the following account data:

- ISX/LSA job CPU time from the time the previous ACCOUNT command was issued (T-TIME),
- the number of LU 6.2 RECEIVE operations from the time the previous ACCOUNT command was issued (RECEIVE),
- the number of inbound frames from the time the previous ACCOUNT command was issued (I-FRAME),
- the number of LU 6.2 SEND operations from the time the previous ACCOUNT command was issued (SEND),
- the number of outbound frames from the time the previous ACCOUNT command was issued (O-FRAME),
- the number of SSCH/SIO instructions for real CTCA from the time the previous ACCOUNT command was issued (SIO),
- the number of retry I/O operations for real CTCA from the time the previous ACCOUNT command was issued (RETRY),

The fourth data line displays the following account data:

- ISX/LSA job CPU time utilization from the time the previous ACCOUNT command was issued (T-TIME),
- LU 6.2 RECEIVE operations frequency from the time the previous ACCOUNT command was issued (RECEIVE),
- inbound frames frequency from the time the previous ACCOUNT command was issued (I-FRAME),

32 OPERATOR COMMAND DESCRIPTION

- LU 6.2 SEND operations frequency from the time the previous ACCOUNT command was issued (SEND),
- outbound frames frequency from the time the previous ACCOUNT command was issued (O-FRAME),
- SSCH/SIO instructions frequency for real CTCA from the time the previous ACCOUNT command was issued (SIO),
- retry I/O operations frequency for real CTCA from the time the previous ACCOUNT command was issued (RETRY).

The ACCOUNT command issued for ISX/LCS displays four data lines.

The first data line displays the following account data:

- total ISX/LCS job CPU time the account function was operative (T-TIME),
- total number of LU 6.2 RECEIVE operations the account function was operative (RECEIVE),
- total number of inbound frames received from the virtual token-ring the account function was operative (I-FRAME),
- total number of LU 6.2 SEND operations the account function was operative (SEND),
- total number of outbound frames sent to the virtual token-ring the account function was operative (O-FRAME),
- total number of SSCH/SIO instructions for real input CTCA the account function was operative (I-SIO),
- total number of SSCH/SIO instructions for real output CTCA the account function was operative (O-SIO).

The second data line displays the following account data:

- total ISX/LCS job CPU time utilization the account function was operative (T-TIME),
- total LU 6.2 RECEIVE operations frequency the account function was operative (RECEIVE),
- total inbound frames frequency the account function was operative (I-FRAME),
- total LU 6.2 SEND operations frequency the account function was operative (SEND),
- total outbound frames frequency the account function was operative (O-FRAME),
- total SSCH/SIO instructions frequency for real input CTCA the account function was operative (I-SIO),
- total SSCH/SIO instructions frequency for real output CTCA the account function was operative (O-SIO).

The third data line displays the following account data:

- ISX/LCS job CPU time from the time the previous ACCOUNT command was issued (T-TIME),
- the number of LU 6.2 RECEIVE operations from the time the previous ACCOUNT command was issued (RECEIVE),
- the number of inbound frames from the time the previous ACCOUNT command was issued (I-FRAME),

- the number of LU 6.2 SEND operations from the time the previous ACCOUNT command was issued (SEND),
- the number of outbound frames from the time the previous ACCOUNT command was issued (O-FRAME),
- the number of SSCH/SIO instructions for real input CTCA from the time the previous ACCOUNT command was issued (I-SIO),
- the number of SSCH/SIO instructions for real output CTCA from the time the previous ACCOUNT command was issued (O-SIO),

The fourth data line displays the following account data:

- ISX/LCS job CPU time utilization from the time the previous ACCOUNT command was issued (T-TIME),
- LU 6.2 RECEIVE operations frequency from the time the previous ACCOUNT command was issued (RECEIVE),
- inbound frames frequency from the time the previous ACCOUNT command was issued (I-FRAME),
- LU 6.2 SEND operations frequency from the time the previous ACCOUNT command was issued (SEND),
- outbound frames frequency from the time the previous ACCOUNT command was issued (O-FRAME),
- SSCH/SIO instructions frequency for real input CTCA from the time the previous ACCOUNT command was issued (I-SIO),
- SSCH/SIO instruction frequency for real output CTCA from the time the previous ACCOUNT command was issued (O-SIO).

ACCOUNT command format:



The following responses can be issued as the result of the ACCOUNT command execution for ISX:

R-TIME	T-TIME	G-TIME	SIE	R-SIO	E-SIO	LCP
Xxxxxx	xxxxxx Xxx%	xxxxxx xxx%	Xxxxxxxx Xxx	xxxxxxx xxx	xxxxxxx xxx	xxxxxx xxx
Xxxxxx	xxxxxx Xxx%	xxxxxx xxx%	Xxxxxxxx Xxx	xxxxxxx xxx	xxxxxxx xxx	xxxxxx xxx

The following responses can be issued as the result of the ACCOUNT command execution for ISX/LSA:

34 OPERATOR COMMAND DESCRIPTION

T-TIME	RECEIVE	I-FRAME	SEND	O-FRAME	SIO	RETRY
XXXXXX	XXXXXX	XXXXXX	XXXXXXXX	XXXXXXXX	XXXXXX	XXXXXX
Xxx%	xxx	xxx	xxx	xxx	xxx	Xxx
XXXXXX	XXXXXX	XXXXXX	XXXXXXXX	XXXXXXXX	XXXXXX	XXXXXX
xxx%	xxx	xxx	xxx	xxx	xxx	Xxx

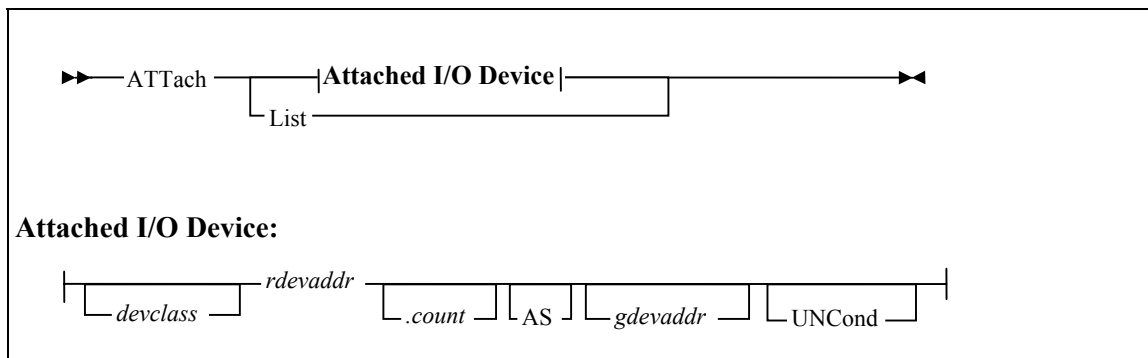
The following responses can be issued as the result of the ACCOUNT command execution for ISX/LCS:

T-TIME	RECEIVE	I-FRAME	SEND	O-FRAME	I-SIO	O-SIO
XXXXXX	XXXXXX	XXXXXX	XXXXXXXX	XXXXXXXX	XXXXXX	XXXXXX
Xxx%	xxx	xxx	xxx	xxx	xxx	xxx
XXXXXX	XXXXXX	XXXXXX	XXXXXXXX	XXXXXXXX	XXXXXX	XXXXXX
Xxx%	xxx	xxx	xxx	xxx	xxx	xxx

ATTACH Command

The ATTACH command is designed to assign a real I/O device to the guest I/O configuration.

ATTACH command format:



devclass

defines a class of the attached I/O device. If *devclass* is not specified, then the device class is received from zOS. The following codes can be specified to define the device class:

DASD

adds a CKD type direct access storage device with the *rdevaddr* real device number to the guest configuration at the device number specified by *gdevaddr*.

F512

adds a FBA type direct access storage device with the *rdevaddr* real device number to the guest configuration at the device number specified by *gdevaddr*.

TAPE

adds a magnetic tape device with the *rdevaddr* real device number to the guest configuration at the device number specified by *gdevaddr*.

UREC

adds a unit record device with the *rdevaddr* real device number to the guest configuration at the device number specified by *gdevaddr*.

GRAF

adds a local display with the *rdevaddr* real device number to the guest configuration at the device number specified by *gdevaddr*.

TERM

adds a telecommunication device with the *rdevaddr* real device number to the guest configuration at the device number specified by *gdevaddr*.

CTCA

adds a channel-to-channel adapter with the *rdevaddr* real device number to the guest configuration at the device number specified by *gdevaddr*.

SPEC

adds a special type device with the *rdevaddr* real device number to the guest configuration at the device number specified by *gdevaddr*.

rdevaddr

is the real device number of the I/O device to be assigned.

.count

is a decimal value designating the number of real devices to be assigned starting with the real device at *rdevaddr* device number.

gdevaddr

is the guest device number that will be assigned to the I/O device. If *gdevaddr* is not specified, the real device number will be assigned to the I/O device. Up to 4095 I/O devices have to be attached or defined for the guest system.

UNCond

defines that the device must be attached even if it is not allocated to the ISX job by zOS.

List

gives the full list of attached real I/O devices and their state:

.gdevaddr(rdevaddr)-s - for each real I/O device, where 's' is one of the following codes:

R - the device is ready for I/O,

N - the device is not operational,

A - the device is doing I/O,

I - I/O interruption is pending for the device.

BEGIN Command

The BEGIN command issued for ISX is designed to continue execution of a guest at either the specified storage address or the location pointed to by the current guest PSW. The BEGIN command issued for ISX/LSA or ISX/LCS is designed to start or continue execution of the emulated token-ring adapter for zOS.

BEGIN command format:



hexaddr

is the hexadecimal storage address where guest execution is to begin.

CAC Command

The CAC command is designed to enter input data for the emulated guest console.

CAC command format:

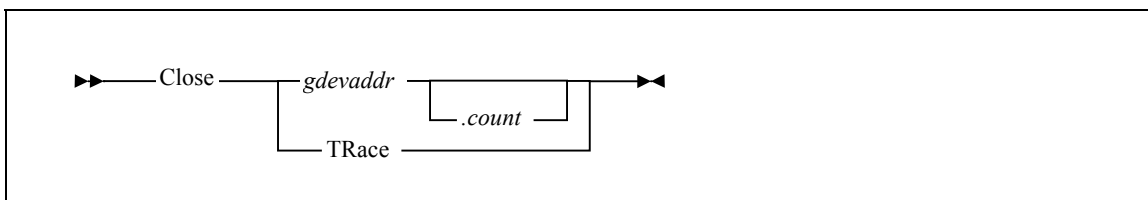
**text**

is the input data for the guest console. If the command is issued, but a read operation is not pending on the guest console, then an “attention” I/O interruption is emulated for the guest console.

CLOSE Command

The CLOSE command is designed to revoke the current linkages between emulated I/O devices and the corresponding zOS SYSOUT data sets, or between active trace processing and the SYSOUT data set used for trace record printing. The data sets are closed and unallocated from the ISX, ISX/LSA, or ISX/LCS job. Once the data sets are freed they are available for processing in zOS.

CLOSE command format:

**gdevaddr**

defines the guest device number of the emulated device for which the current linkage with the corresponding zOS SYSOUT data set is to be revoked. If the guest device is emulated token-ring adapter and its message log is active, then a new SYSOUT data set is created for message log of the device. If the guest device is an emulated card punch or printer associated with a SYSOUT data set, then a new SYSOUT data set is created for emulation of the devices.

.count

is a decimal value designating the number of emulated devices to be closed starting with the guest device at *gdevaddr* device number.

TRace

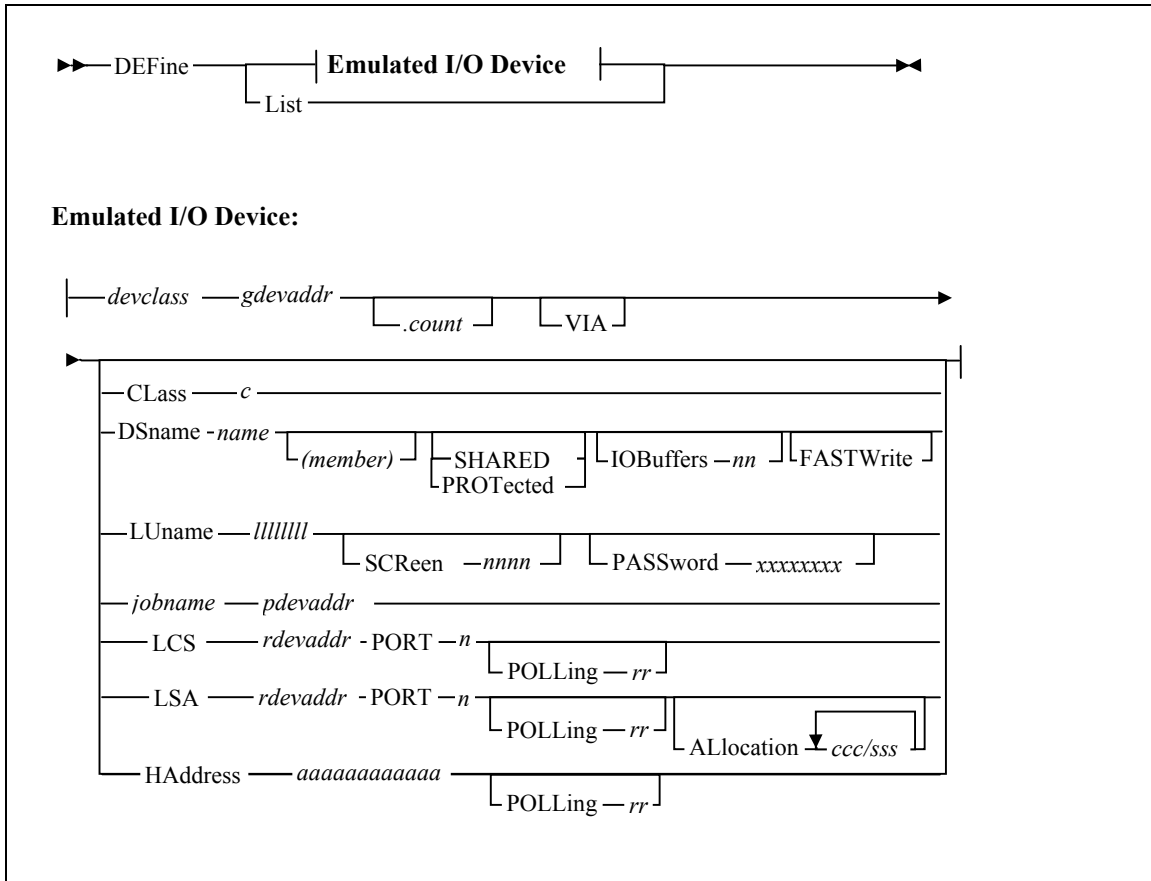
revokes current linkage with the SYSOUT data set used to print traced records.

Note. For emulated devices SYSOUT data sets are created with the DEST parameter equal to Gxxxxnnn, where nnn is a data set sequence number for emulated device xxxx. For trace data sets the DEST parameter equates to TRACEnnn, where nnn is the data set sequence number.

DEFINE Command

The DEFINE command of ISX is designed to define an emulated I/O device for the guest system. The DEFINE command of ISX/LSA is designed to define an emulated LSA adapter for zOS. The DEFINE command of ISX/LCS is designed to define an emulated LCS adapter for zOS.

DEFINE command format for ISX:



devclass

defines a class of the defined I/O device. The following codes can be specified to define the device class:

CON or 3215

adds an emulated Console Printer Keyboard (looks like 3215) to the guest configuration at the device number specified by *gdevaddr*.

RDR or 2540R

adds an emulated card reader device (looks like 2540R) to the guest configuration at the device number specified by *gdevaddr*.

PUN or 2540P

adds an emulated card punch device (looks like 2540P) to the guest configuration at the device number specified by *gdevaddr*.

38 OPERATOR COMMAND DESCRIPTION

PRT or 1403

adds an emulated printer device (looks like 1403) to the guest configuration at the device number specified by *gdevaddr*.

FBA or 9332 or 9335 or 9336

adds an emulated FBA DASD device (looks like 9332, 9335, or 9336) to the guest configuration at the device number specified by *gdevaddr*.

3380 or 3390

adds an emulated CKD DASD device (looks like 3380 or 3390) to the guest configuration at the device number specified by *gdevaddr*.

3480 or 3490

adds an emulated magnetic tape device (looks like 3480 or 3490) to the guest configuration at the device number specified by *gdevaddr*.

GRF or 3270

adds an emulated local non-SNA display unit (looks like a 3274 Model 1D with Model 1, 2, 3, 4 or 5 devices without QUERY capability with the maximum screen size has to be defined by the SCREEN operand) to the guest configuration at the device number specified by the *gdevaddr* operand.

CCA or 3088

adds an emulated channel-to-channel adapter to the guest I/O configuration at the device number specified by the *gdevaddr* operand.

CET

adds an emulated token-ring adapter to the guest I/O configuration at the device numbers beginning from *gdevaddr*. The emulated token-ring adapter looks like the IBM Token-Ring Subsystem (IEEE 802.5) of the IBM 9221 Processors.

LCS

adds an emulated token-ring adapter to the guest I/O configuration at the device numbers beginning from *gdevaddr*. The emulated token-ring adapter looks like the LCS Interface of a processor attached platform such as OSA or a channel attached platform as the 3172.

gdevaddr

is the guest device number that will be assigned to the I/O device. For an emulated CET adapter *gdevaddr*, *gdevaddr+1*, *gdevaddr+2*, and *gdevaddr+3* will be assigned. For an emulated LCS adapter *gdevaddr* and *gdevaddr+1* will be assigned. Up to 4095 I/O devices have to be defined or attached for the guest system.

.count

is a decimal value designating the number of emulated devices to be defined starting with the guest device at *gdevaddr* device number.

Class *c*

defines the SYSOUT class of data sets associated with an emulated PRT or PUN device.

DSname name [(member)]

defines the name of the sequential data set or member name and name of a partitioned data set to be used during emulation of a card reader device, a card punch device, or a printer. Defines the name of a VSAM data set to be used for emulation of a FBA or CKD DASD device, or a magnetic tape device. If it is the data set for a card punch, card reader, or printer device emulation, then the data set has to have one of the following record formats: F, FM, FB, FBM, V, VM, VB, VBM, U, or UM. For card punch device emulation the logical record length (block size for format U) must not be less than 80 bytes (81 bytes for character) for the record format of F and not less than 84 bytes (85 bytes for the record format with machine character) for the record format of V. For a printer emulation its logical record length (block size for format U) has to be not less than 132 bytes (133 bytes for the record format with machine character) for the record format of F and not less than 136 bytes (137 bytes for the record format with machine character) for the record format of V. For FBA DASD emulation the VSAM data set has to be a fixed-length RRDS type with a record length of 512 bytes. For magnetic tape emulation the VSAM data set has to be a fixed-length RRDS type with a record length of 4096 bytes. For CKD DASD emulation the VSAM data set has to be a fixed-length RRDS type with a record length not less than 3104 bytes. The recommended record length is 25542 bytes for emulated 3380 and 30211 bytes for emulated 3390.

SHARED

defines that the emulated DASD device has to be shared. If the operand is specified, then the corresponding VSAM data set is allocated with DISP=SHR and reserved for shared use among ISX jobs in zOS. If it is not specified, then the data set is allocated with DISP=OLD and reserved for exclusive use among ISX jobs in zOS.

PROTECTED

defines that the emulated magnetic tape device has to be write protected. If the operand is specified, then the corresponding VSAM data set is allocated with DISP=SHR. If it is not specified, then it is allocated with DISP=OLD.

IOBuffers nn

defines that nn I/O buffers have to be used for the emulated FBA DASD, CKD DASD, or magnetic tape. Maximal value for the number of I/O buffers is 99. For FBA DASD and CKD DASD the default value is 16, for a magnetic tape the default value is 8.

FASTWrite

defines that for emulated DASD or magnetic tape device fast write operations are performed instead of usual write operations. In fast write mode the corresponding device emulator return device end status to the guest system when the data is stored in I/O buffer. The write to the data set is transparent to the guest system, but, if any, the data remains in I/O buffer no more than two seconds.

LUname /llllll

defines the logical unit name that will be used to connect the guest device *gdevaddr* with a VTAM logical unit. The VTAM logical unit has to be LU type 0 or 2. If the logical unit name is defined as LOGON, then the emulated device can be connected with a real VTAM terminal via the LOGON procedure of ISX.

SCReen nnnn

defines the maximum screen size of the emulated display unit. The following values are allowed:

- 480 - for 12X40 screen
- 1920 - for 24X80 screen
- 2560 - for 32X80 screen

40 OPERATOR COMMAND DESCRIPTION

3440 - for 43X80 screen
3564 - for 27X132 screen
The default value is 1920.

PASSword

defines the customer's password for connection with the emulated display when the session is initiated from a VTAM terminal.

jobname

defines ISX job name for the partner side of the emulated channel-to-channel adapter.

pdevaddr

defines the guest device number for the partner side of the emulated channel-to-channel adapter. If the device number is not defined, then the *gdevaddr* device number is assumed.

LCS *rdevaddr*

defines the real device numbers *rdevaddr* and *rdevaddr*+1 of IBM non-EMIF LAN Channel Station (LCS) devices of OSA/1 or OSA/2 adapter to be allocated to the ISX job for emulation of the guest IBM Token-Ring Subsystem (IEEE 802.5) of the IBM 9221 Processors.

LSA *rdevaddr*

defines the real device number *rdevaddr* of the IBM Link Services Architecture (LSA) device of OSA/2 adapter or a channel attached device such as 3172 to be allocated to the ISX job for emulation of the guest IBM Token-Ring Subsystem (IEEE 802.5) of the IBM 9221 Processors.

If neither LCS *rdevaddr* nor LSA *rdevaddr* value is specified, but the HADDRESS *aaaaaaaaaaaa* operand is specified, then full emulated IBM Token-Ring Subsystem (IEEE 802.5) of the IBM 9221 Processors (CET adapter) or a LCS Interface of a processor attached platform such as OSA or a channel attached platform as the 3172 (LCS adapter) is assumed for connection with the virtual token-ring.

PORT *n*

defines the port *n* of a real token-ring adapter to be used by the ISX job for emulation of the guest token-ring adapter.

HAddress *aaaaaaaaaaaa*

defines the hardware address of the full emulated token-ring adapter. The guest software can change this address. This operand must be specified for emulated LCS adapter.

POLLing *rr*

defines polling time interval *rr* to read inbound/outbound request parameters of the guest system. Valid *rr* values are the following:

- 32 - to read parameters every 32 milliseconds
- 16 - to read parameters every 16 milliseconds
- 8 - to read parameters every 8 milliseconds
- 4 - to read parameters every 4 milliseconds
- 2 - to read parameters every 2 milliseconds
- 1 - to read parameters every 1 millisecond

The default *rr* value is 32.

Allocation *ccc/sss*

defines preallocation of *ccc* connections for service access point *sss* of the guest software for the real LSA adapter. If preallocation is not defined for a service access point, then the average number of currently available connections will be allocated.

List

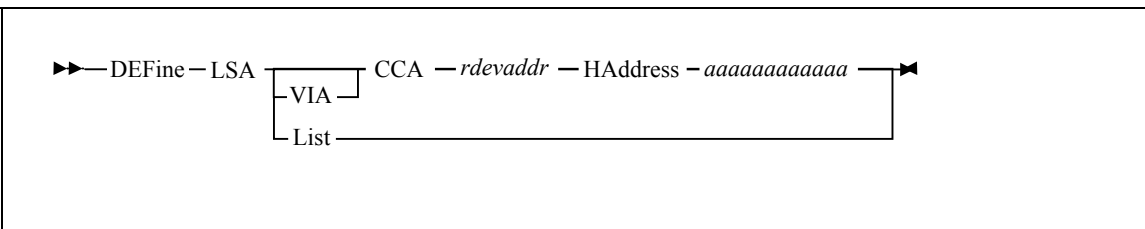
gives the full list of defined emulated I/O devices and their state:

- . *gdevaddr*(3215)-s - for emulated console,
- . *gdevaddr*(2540R)-s - for each emulated card reader,
- . *gdevaddr*(2540P)-s - for each emulated card punch,
- . *gdevaddr*(1403)-s - for each emulated printer,
- . *gdevaddr*(9332/9335/9336)-s - for each emulated FBA DASD,
- . *gdevaddr*(3270)-s - for each emulated 3270,
- . *gdevaddr*(3380/3390)-s - for each emulated CKD DASD,
- . *gdevaddr*(3480/3490)-s - for each emulated magnetic tape,
- . *gdevaddr*(3080)-s - for each emulated channel-to-channel adapter,
- . *gdevaddr*(CETI)-s - for each emulated CET adapter,
- . *gdevaddr*(LCS)-s - for each emulated LCS adapter,

where 's' is one of the following codes:

- R - the device is ready for I/O,
- N - the device is not operational,
- A - the device is doing I/O,
- I - I/O interruption is pending for the device.

DEFINE command format for ISX/LSA:



LSA

adds an emulated token-ring adapter for zOS. The emulated token-ring adapter looks like the LSA Interface of a processor attached platform such as OSA or a channel attached platform such as the 3172.

CCA *rdevaddr*

defines the device number of the real channel-to-channel adapter to be allocated for emulation of the token-ring adapter. The opposite side of the channel-to-channel adapter has to be used by zOS as the real LSA adapter.

42 OPERATOR COMMAND DESCRIPTION

HAddress *aaaaaaaaaaaa*

defines the hardware address of the emulated LSA adapter.

List

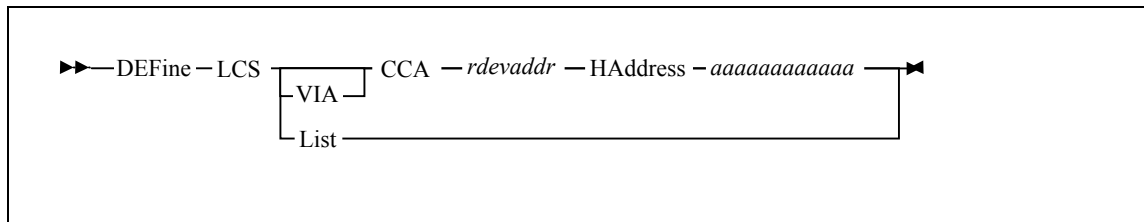
gives state of real CTCA:

. *rdevaddr*(3088) - s

where 's' is one of the following codes:

- R - the CTCA is ready for I/O,
- A - the CTCA is doing I/O,
- I - I/O interruption is pending for CTCA.

DEFINE command format for ISX/LCS:



LCS

adds an emulated token-ring adapter for zOS. The emulated token-ring adapter looks like the LCS Interface of a processor attached platform such as OSA or a channel attached platform such as the 3172.

CCA *rdevaddr*

defines the device numbers *rdevaddr* and *rdevaddr+1* of the real channel-to-channel adapters to be allocated for emulation of the LCS adapter. The opposite sides of the channel-to-channel adapters have to be used by zOS as the real LCS adapter.

HAddress *aaaaaaaaaaaa*

defines the hardware address of the emulated LCS adapter.

List

gives state of real CTCAs:

. *rdevaddr*(3088) - s

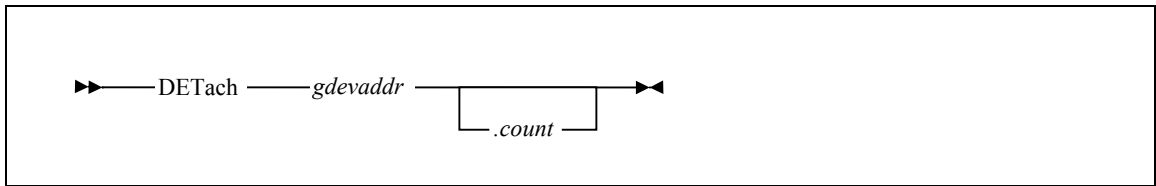
where 's' is one of the following codes:

- R - the CTCA is ready for I/O,
- A - the CTCA is doing I/O,
- I - I/O interruption is pending for CTCA.

DETACH Command

The DETACH command is designed to remove a device from the ISX guest or from the ISX/LSA or ISX/LCS I/O configuration.

DETACH command format:



gdevaddr

is the guest device number of the I/O device to be removed from the guest I/O configuration or the real device number of the channel-to-channel adapter assigned to ISX/LSA or ISX/LCS.

.count

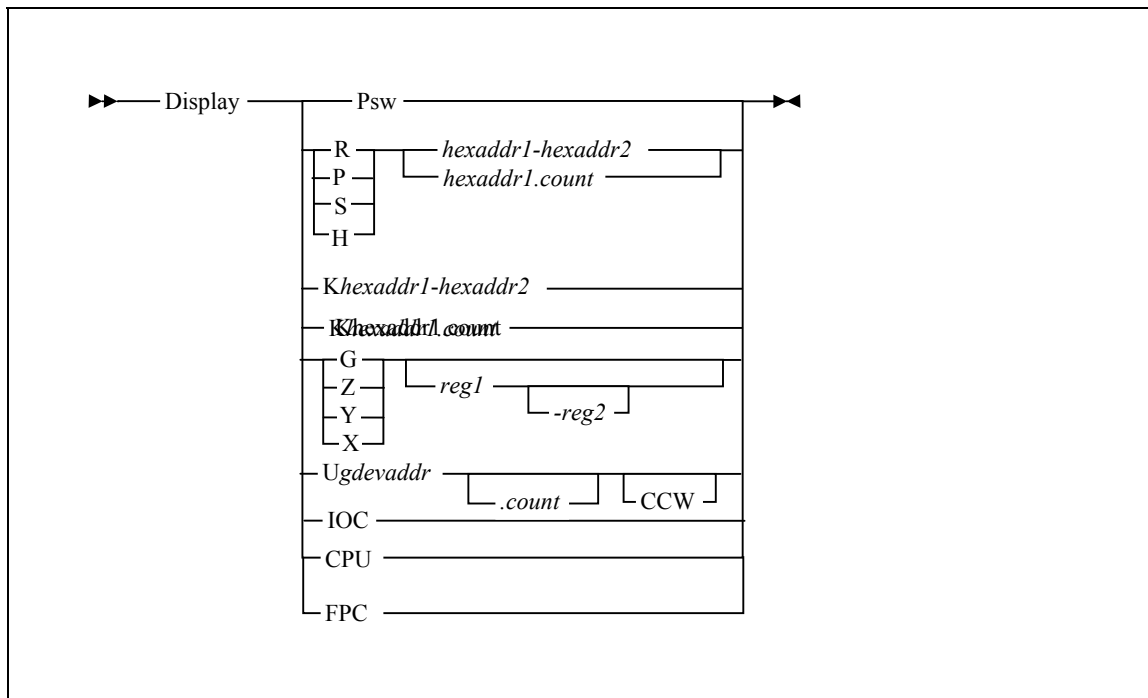
is a decimal value designating the number of devices to be detached starting with the guest device at *gdevaddr* device number.

DISPLAY Command

The DISPLAY command is designed to display the following data from the guest system environment:

- Interpretive space storage (guest real storage) contents.
- Primary virtual storage contents.
- Secondary virtual storage contents.
- Home virtual storage contents.
- Interpretive space storage protection keys.
- Guest general registers.
- Guest access registers.
- Guest Floating-point registers.
- Guest floating-point control register.
- Guest control registers.
- Guest program status word.
- Guest I/O device(s) states.
- Guest I/O configuration.
- Guest CPU state.

DISPLAY command format:



R, P, S or H

defines the guest storage to be displayed. 'R' defines the guest real storage to be displayed. 'P' defines the current guest primary virtual storage to be displayed. 'S' defines the current guest secondary virtual storage to be displayed. For ISX 'H' defines the current guest home virtual storage to be displayed.

Note. If neither 'R' nor 'P' nor 'S' nor 'H' is specified, then the guest storage type is defined via the guest PSW and the guest control registers.

hexaddr1

is the hexadecimal location address that is to be displayed. The specified storage contents are displayed in hexadecimal and with EBCDIC translation. If the prefix **R**, **P**, **S** or **H** is not specified, then the currently visible guest storage (real, primary virtual, secondary virtual or home) is displayed.

Khexaddr1

is the hexadecimal guest address that is to be used to display the corresponding storage protection key in hexadecimal.

-hexaddr2

is the last of the range of hexadecimal location addresses, whose contents are to be displayed or whose storage protection key are to be displayed.

.count

is a hexadecimal value designating the number of bytes of storage, whose contents are to be displayed or whose storage protection key are to be displayed starting with the byte at *hexaddr1*.

G[*reg1*[-*reg2*]]

displays the contents of the general registers. If **G** is specified without a register number, the contents of all general registers are to be displayed. If only *reg1* is specified, then only *reg1* contents are to be displayed. If both *reg1* and *reg2* are specified, then the contents of all general registers from *reg1* to *reg2* are to be displayed.

Z[*reg1*[-*reg2*]]

for ISX/390 displays the contents of the access registers. If **Z** is specified without a register number, the contents of all access registers are to be displayed. If only *reg1* is specified, then only *reg1* contents are to be displayed. If both *reg1* and *reg2* are specified, then the contents of all registers from *reg1* to *reg2* are to be displayed.

Y[*reg1*[-*reg2*]]

displays the contents of the floating-point registers. If **Y** is specified without a register number, the contents of all floating-point registers are to be displayed. If only *reg1* is specified, then only *reg1* contents are to be displayed. If both *reg1* and *reg2* are specified, then the contents of all floating-point registers from *reg1* to *reg2* are to be displayed.

X[*reg1*[-*reg2*]]

displays the contents of the control registers. If **X** is specified without a register number, the contents of all control registers are to be displayed. If only *reg1* is specified, then only *reg1* contents are to be displayed. If both *reg1* and *reg2* are specified, then contents of all control registers from *reg1* to *reg2* are to be displayed.

Ugdevaddr[.count] [CCW]

displays messages about the states of guest I/O device(s) and currently executed I/O operations beginning from the device with *gdevaddr* device number. If *count* is specified, then messages are displayed for the specified number of devices. If **CCW** is specified, then the last executed guest channel program is displayed.

FPC

displays the guest floating-point control register.

IOC

displays the guest I/O configuration.

CPU

displays real and guest CPU identifiers, guest prefix, machine check interruptions pending, and external interruptions pending.

Psw

displays the current guest Program Status Word.

Note. An empty input command line is treated as a request to stop the **DISPLAY** command execution.

DUMP Command

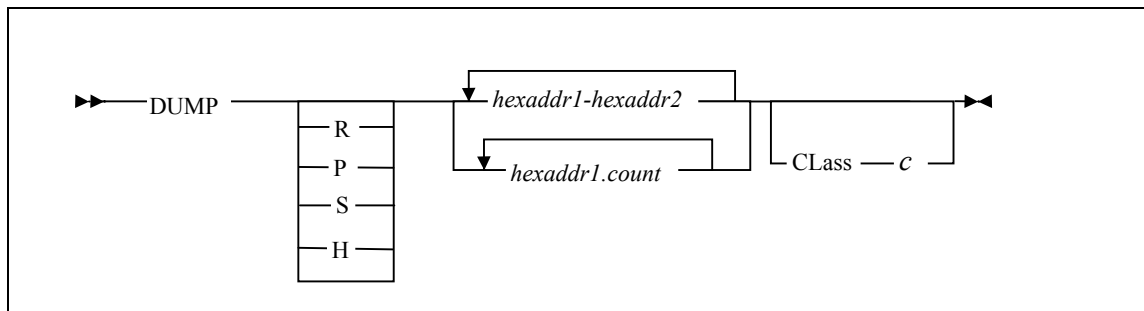
The **DUMP** command is designed to produce a dump of the guest system environment including:

46 OPERATOR COMMAND DESCRIPTION

- Guest (real, current primary virtual, current secondary virtual or current home virtual) storage locations and its protection keys.
- Guest general registers.
- Guest access registers.
- Guest Floating-point registers.
- Guest floating-point control register.
- Guest control registers.
- Guest program status word.
- All parameters established by the SET command for the guest interpretive space.

A special SYSOUT data set is dynamically created by ISX to contain the dump.

DUMP command format:



R, P, S or H

defines the guest storage to be dumped. '**R**' defines the guest real storage to be dumped. '**P**' defines the current guest primary virtual storage to be dumped. '**S**' defines the current guest secondary virtual storage to be dumped. For ISX '**H**' defines the current guest home virtual storage to be dumped.

Note. If neither '**R**' nor '**P**' nor '**S**' nor '**H**' is specified, then the guest storage type is defined via the guest PSW and the guest control registers.

hexaddr1

is the hexadecimal location address that is to be dumped. The storage contents are dumped in hexadecimal and with EBCDIC translation.

-hexaddr2

is the last of the range of hexadecimal location addresses, whose contents are to be dumped.

.count

is a hexadecimal value designating the number of bytes of storage to be dumped starting with the byte at *hexaddr1*.

Class c

defines the SYSOUT class of data sets associated with the dump. The default value is D.

EXTERNAL Command

The EXTERNAL command is designed to simulate an external key interruption to the guest system.

EXTERNAL command format:



HMC Command

The HMC command is designed to enter input data for the emulated hardware management console.

HMC command format:



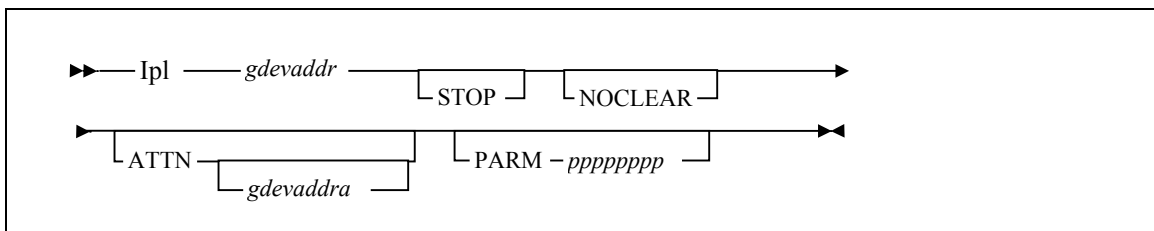
text

is the input data for the emulated hardware management console.

IPL Command

The IPL command is designed to simulate an initial program load of the guest system.

IPL command format:



gdevaddr

is the guest device number of the I/O device that contains the guest loader. The device has to be a unit record, DASD, or magnetic tape one.

STOP

halts the guest during the IPL just before the initial PSW is loaded.

NOCLEAR

defines that the guest environment is not to be clear reset and the CPU status is to be stored before initial program loading.

ATTN [*gdevaddr*]

during the IPL generates an “attention” I/O interruption from the specified guest device with the device number *gdevaddr*. If the device number is not specified, then the I/O interruption is generated from the guest console.

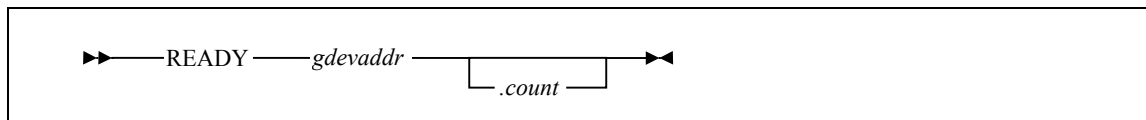
PARM *pppppppp*

defines up to eight characters of data to be available for the IPL'ed guest system as load parameters.

READY Command

The READY command is designed to examine and set the operational status of the specified real device.

READY command format:



gdevaddr

defines that I/O device with the guest device number of *gdevaddr* to be examined and set up.

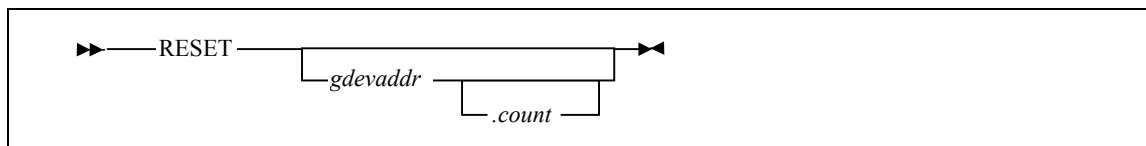
.count

is a decimal value designating the number of devices to be examined starting with the guest device at *gdevaddr* device number.

RESET Command

The RESET command is designed to reset the guest environment or an I/O device assigned to the guest.

RESET command format:



gdevaddr

defines that the I/O device with the guest device number of *gdevaddr* is to be reset. If the device is an emulated one and is linked with a SYSOUT data set this linkage is to be revoked, the data set is closed and unallocated. If *gdevaddr* is not specified then the guest environment and all guest I/O devices are to be reset and all device linkages with SYSOUT data sets are to be revoked.

.count

is a decimal value designating the number of devices to be reset starting with the guest device at *gdevaddr* device number.

RESTART Command

The RESTART command is designed to restart the guest system. When this command is entered the current guest PSW is saved at the guest storage address of X'08' and the guest resumes execution at the address in the PSW loaded from the interpretive space storage location X'00'.

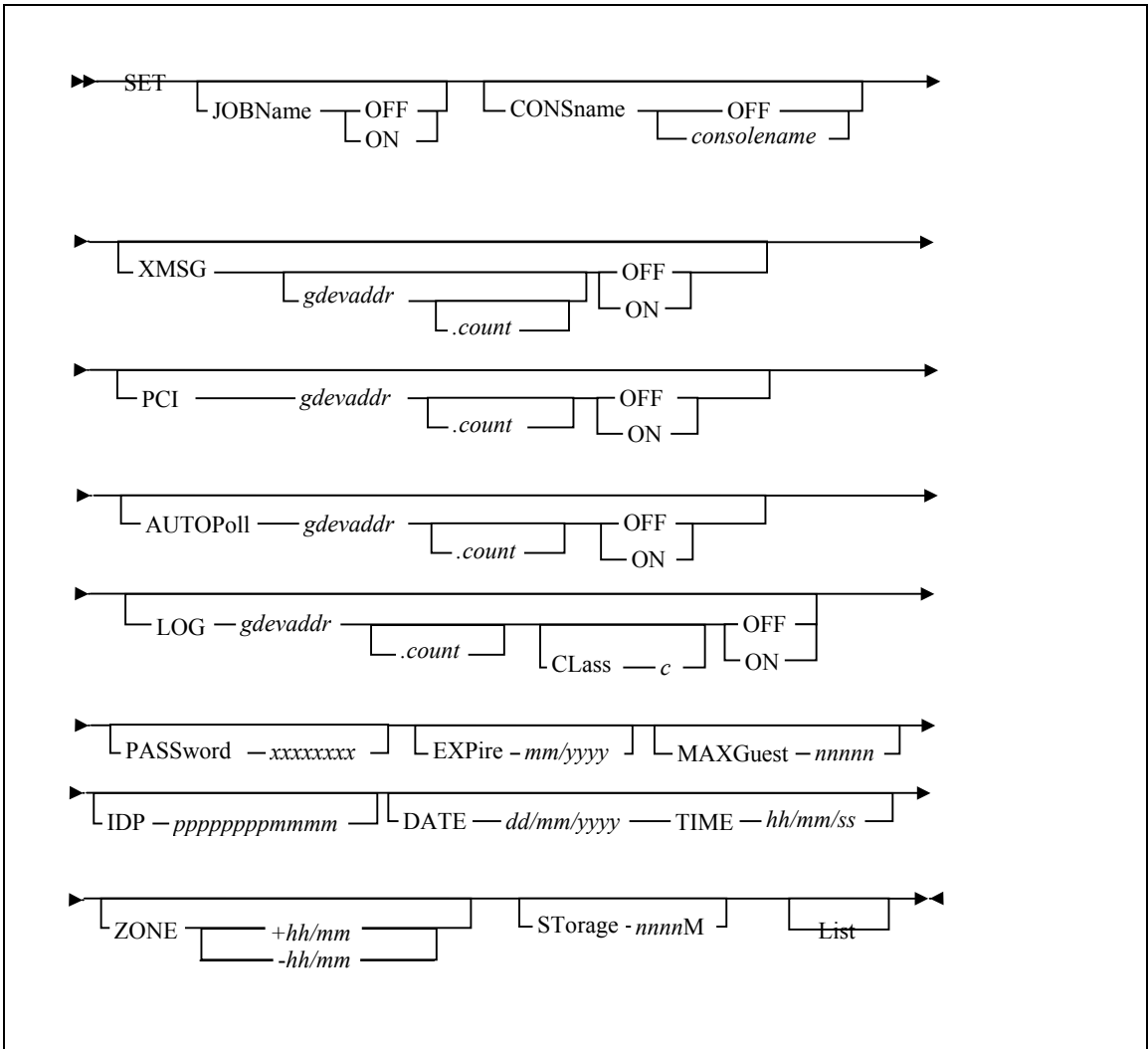
RESTART command format:



SET Command

The SET command is designed to define the functional parameters of ISX, ISX/LSA, and ISX/LCS.

SET command format for ISX:



AUTOPoll

controls whether or not ISX tests each BTAM autopoll CCW to see if it has been dynamically modified. The operand *gdevaddr* has to be specified to define the corresponding guest device numbers for which this parameter is applicable.

PCI

controls whether or not ISX tests each PCI interruption is reflected to the guest system to see if the channel program has been dynamically modified. The operand *gdevaddr* has to be specified to define the corresponding guest device numbers for which this parameter is applicable.

JOBName

controls whether or not ISX displays the corresponding job name before all messages to the operator.

XMSG

controls whether or not ISX issues all messages about operation and I/O exceptions.

CONSname *consolename*

defines the name of the zOS console to be used to communicate with ISX and with emulated printer keyboard of a guest system.

LOG

for ISX controls whether or not ISX writes all inbound/outbound frames of the token-ring and all inbound/outbound messages of the emulated token-ring adapter in the log data set. In this case the operand *gdevaddr* has to be specified to define the corresponding guest device numbers to which this parameter is applicable. When the log data set is filled with 20.000 records it is automatically closed, unallocated, and new data set is allocated, opened, and used to contain subsequent records.

gdevaddr

defines that AUTOPOLL, XMSG, PCI, or LOG functions are applicable to the guest devices beginning from the guest device with the device number of *gdevaddr*.

.count

is a decimal value designating the number of devices starting with the device number of *gdevaddr* for which AUTOPOLL, XMSG, PCI, or LOG functions are applicable.

Class *c*

defines the SYSOUT class '*c*' of data sets associated with the message log of the emulated token-ring adapter. The default value is L.

ON

indicates that the AUTOPOLL, JOBNAME, XMSG, PCI, LOG function is activated.

OFF

indicates that the AUTOPOLL, JOBNAME, XMSG, PCI, LOG, CONSNAME function is deactivated.

STorage *nnnnM*

defines the interpretive space size in virtual fixed storage of a region as *nnnnM*, where *nnnn* is a value from 1 to 64 for S/370 architecture and from 1 to 2047 for S/390 architecture.

IDP *ppppppppmmmm*

defines the CPU *pppppppp* and model *mmmm* identifiers for the guest system.

DATE *dd/mm/yyyy*

defines the initial date for the interpretive space in the form of *dd/mm/yyyy*, where *dd* is day number, *mm* is month number, and *yyyy* is year number.

TIME *hh/mm/ss*

defines the initial clock value for the interpretive space in the form of *hh/mm/ss*, where *hh* is hour number, *mm* is minute number, and *ss* is second number.

ZONE [-|+]*hh/mm*

defines the time zone value in the form of *hh/mm*, where *hh* is number of hours, *mm* is number of minutes. The time zone value is added to the zOS current time to define the local time for a guest system.

PASSword

defines the customer's password for usage of ISX. The value depends on CPU ID of the real computer, EXPIRE, and MAXGUEST values.

EXPIre

defines expiration date of the ISX license. If the operand is not defined, then the ISX license is unlimited. In any case the value depends on customer's password.

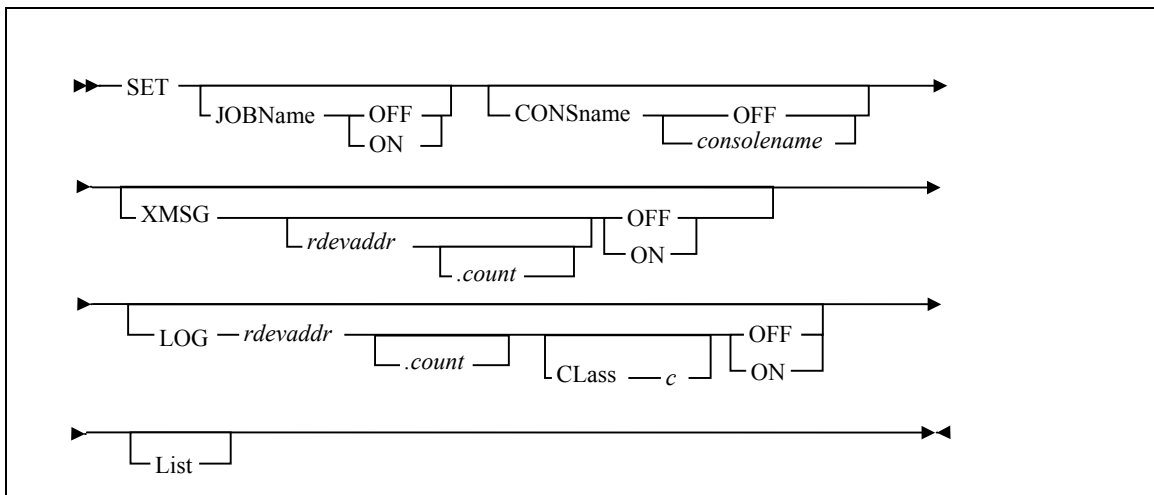
MAXGuest

defines maximal number of guest systems executed by ISX. If the operand is not defined, then the number of executed guest systems is unlimited. In any case the value depends on customer's password.

List

gives the full list of functional parameters set for ISX.

SET command format for ISX/LSA and ISX/LCS:



JOBName

controls whether or not ISX/LSA or ISX/LCS displays the corresponding job name before all messages to the operator.

XMSG

controls whether or not ISX/LSA or ISX/LCS issues all messages about operation and I/O exceptions.

CONSname *consolename*

defines the name of the zOS console to be used to communicate with ISX/LSA or with ISX/LCS.

LOG

controls whether or not the emulator writes all inbound/outbound frames of the virtual token-ring and all inbound/outbound primitives of zOS in the log data set. In this case the operand *rdevaddr* has to define the channel-to-channel adapter specified in the DEFINE command for ISX/LSA or ISX/LCS. When the log data set is filled with 20.000 records it is automatically closed, unallocated, and new data set is allocated, opened, and used to contain subsequent records.

rdevaddr

defines that XMSG or LOG functions are applicable to the real devices beginning from the device with the device number of *rdevaddr*.

.count

is a decimal value designating the number of devices starting with the device number of *gdevaddr* for which XMSG or LOG functions are applicable.

Class *c*

defines the SYSOUT class '*c*' of data sets associated with the message log of the emulated token-ring adapter. The default value is L.

ON

indicates that the JOBNAME, XMSG, LOG function is activated.

OFF

indicates that the JOBNAME, XMSG, LOG, CONSNAME function is deactivated.

List

gives the full list of functional parameters set for ISX/LSA or ISX/LCS.

SHUTDOWN Command

The SHUTDOWN command is designed to cancel all ISX operations for the guest system or ISX/LSA or ISX/LCS operation for an emulated token-ring adapter.

SHUTDOWN command format:



STOP Command

The STOP command is designed to halt guest execution processing of ISX or token-ring emulation processing of ISX/LSA or ISX/LCS.

STOP command format:



hexaddr

is the hexadecimal value that defines the address in the current guest PSW to halt guest execution processing. If *hexaddr* value is not specified, then the current guest address is assumed.

OFF

cancels a previously issued request to halt the guest at the specified address.

Note. An empty input command line is treated as the STOP command without an operand.

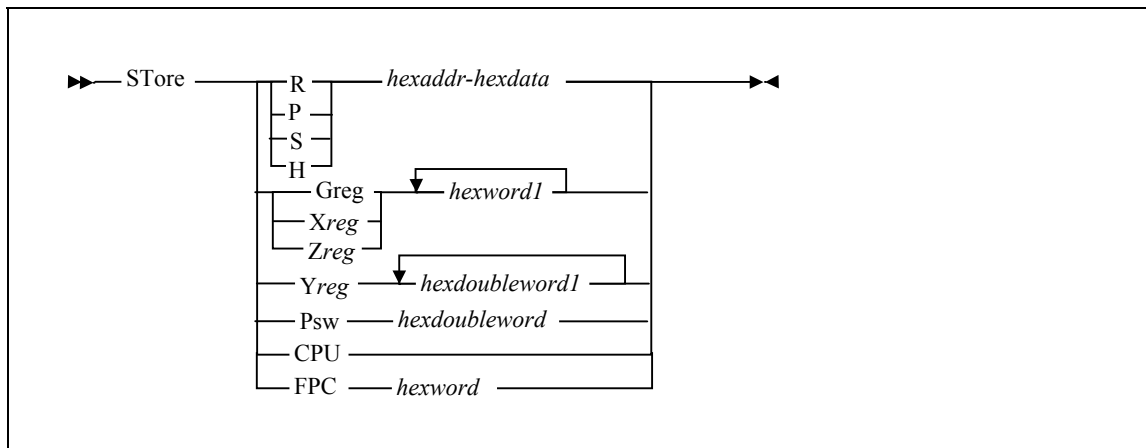
STORE Command

The STORE command is designed to alter the following guest environment:

- Interpretive space storage (guest real storage) contents.
- Primary virtual storage contents.
- Secondary virtual storage contents.
- Home virtual storage contents.
- Guest general registers.
- Guest access registers.
- Guest floating-point registers.
- Guest floating-point control register.
- Guest control registers.
- Guest program status word.

The STORE command allows the storage of the guest CPU status.

STORE command format:



R, P, S or H

defines the guest storage to be altered. '**R**' defines the guest real storage to be altered. '**P**' defines the current guest primary virtual storage to be altered. '**S**' defines the current guest secondary virtual storage to be altered. For ISX '**H**' defines the current guest home virtual storage to be altered.

Note. If neither '**R**' nor '**P**' nor '**S**' nor '**H**' is specified, then the guest storage type is defined via the guest PSW and the guest control registers.

hexaddr

is the hexadecimal location address that is to be altered. If the prefix **R**, **P**, **S** or **H** is not specified, then the currently visible guest storage (real, primary virtual, secondary virtual or home) is to be altered.

hexdata

stores the hexadecimal data (hexdata) starting at the address specified by *hexaddr* hexadecimal value.

Greg hexword1

stores the hexadecimal word(s) of data in successive general registers starting at the register specified by *reg*.

Zreg hexword1

for ISX/390 stores the hexadecimal word(s) of data in successive access registers starting at the register specified by *reg*.

Xreg hexword1

stores the hexadecimal word(s) of data in successive control registers starting at the register specified by *reg*.

Yreg hexdoubleword1

stores the hexadecimal doubleword(s) of data in successive floating-point registers starting at the register specified by *reg*.

Psw hexdoubleword

stores the hexadecimal doubleword of data in the guest Program Status Word.

FPC hexword

stores the hexadecimal word of data in the guest floating-point control register.

CPU

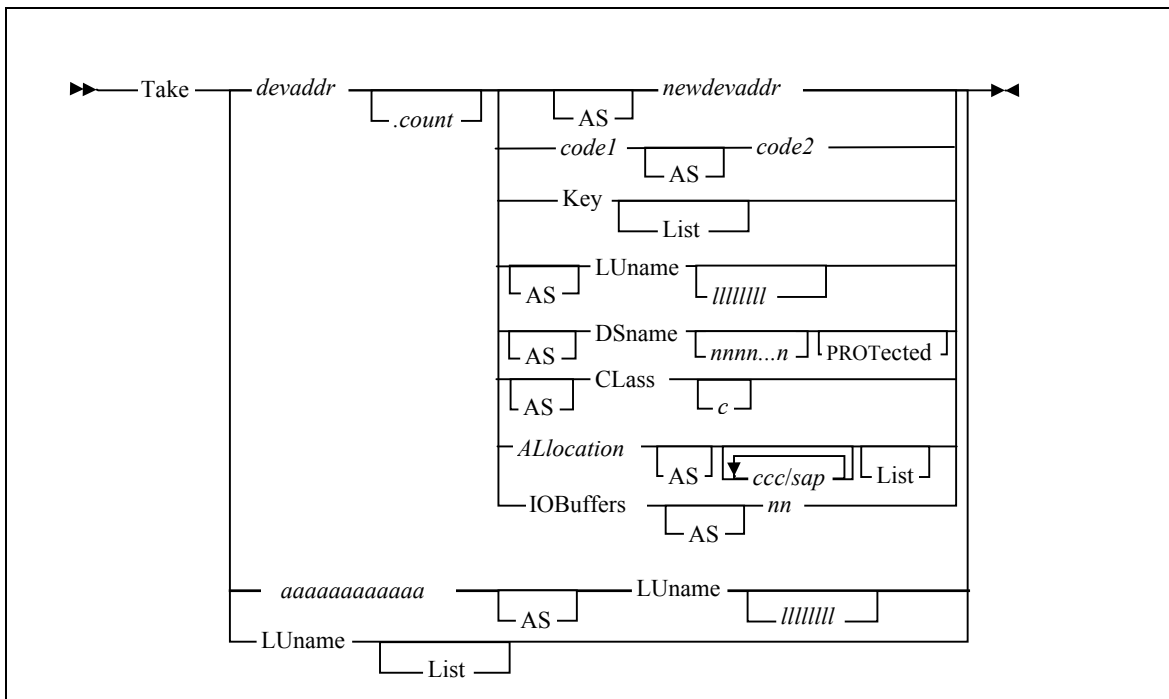
stores the guest CPU status (CPU timer value, clock comparator value, current PSW, prefix value, access registers contents, floating point registers contents, general registers contents, and control registers contents) in corresponding assigned storage locations of the guest storage.

TAKE Command

The TAKE command is designed to redefine device parameters that were previously defined via the ATTACH, DEFINE, or TAKE command.

For all guest devices the TAKE command allows the redefinition of guest device numbers. For emulated 3270 display units the TAKE command allows the redefinition of function keys and logical unit names for the corresponding real devices. For emulated card reader, card punch, or printer devices the TAKE command allows the device to be assigned to a different data set name. For emulated card punch or printer the TAKE command allows the I/O device to be assigned to a different SYSOUT class. For emulated FDA DASD, CKD DASD, or magnetic tape the TAKE command allows to redefine the number of I/O buffers have to be used for VSAM I/O during the I/O device emulation. For emulated token-ring adapters the TAKE command allows the redefinition of the desired number of connections to specified SAPs or to associate VTAM LU names with the destination hardware address of virtual token-ring adapter.

TAKE command format for ISX:



devaddr

defines that the characteristics of the guest I/O device with the guest device number *devaddr* that are to be redefined.

.count

is a decimal value designating the number of devices to be redefined starting with the guest device at *devaddr* device number.

aaaaaaaaaaaa

defines destination hardware address of emulated token-ring adapter that has to be associated with the specified LU name.

newdevaddr

specifies the guest I/O device with the guest device number of *devaddr* is to have a new guest device number of *newdevaddr*.

code1

the following codes can be specified for functional keys that have to be redefined for emulated 3270 display unit(s):

- ENTER** - enter function key
- PF1** - program function 1 key
- PF2** - program function 2 key
- PF3** - program function 3 key
- PF4** - program function 4 key
- PF5** - program function 5 key
- PF6** - program function 6 key
- PF7** - program function 7 key
- PF8** - program function 8 key
- PF9** - program function 9 key
- PF10** - program function 10 key
- PF11** - program function 11 key
- PF12** - program function 12 key
- PF13** - program function 13 key
- PF14** - program function 14 key
- PF15** - program function 15 key
- PF16** - program function 16 key
- PF17** - program function 17 key
- PF18** - program function 18 key
- PF19** - program function 19 key
- PF20** - program function 20 key
- PF21** - program function 21 key
- PF22** - program function 22 key
- PF23** - program function 23 key
- PF24** - program function 24 key
- PA1** - program attention 1 key
- PA2** - program attention 2 key
- PA3** - program attention 3 key

code2

the following codes can be specified to define the new functions for *code1* keys:

- ENTER** - enter function key
- PF1** - program function 1 key
- PF2** - program function 2 key
- PF3** - program function 3 key
- PF4** - program function 4 key
- PF5** - program function 5 key

PF6	- program function 6 key
PF7	- program function 7 key
PF8	- program function 8 key
PF9	- program function 9 key
PF10	- program function 10 key
PF11	- program function 11 key
PF12	- program function 12 key
PF13	- program function 13 key
PF14	- program function 14 key
PF15	- program function 15 key
PF16	- program function 16 key
PF17	- program function 17 key
PF18	- program function 18 key
PF19	- program function 19 key
PF20	- program function 20 key
PF21	- program function 21 key
PF22	- program function 22 key
PF23	- program function 23 key
PF24	- program function 24 key
PA1	- program attention 1 key
PA2	- program attention 2 key
PA3	- program attention 3 key
TESTREQ	- test request function key
SYSREQ	- system request function key
SWITCH	- terminal switch key
LEAVE	- terminal disconnect key
BREAK	- terminal power off key

The terminal switch function allows the real terminal used for emulation of the guest device *devaddr* to switch to the next emulated guest device with the same logical unit name that was defined with a DEFINE or a TAKE command.

The terminal disconnect function requests that a session with the emulated guest I/O device *devaddr* be ended.

Key List

gives the full list of functional key definitions for emulated display units.

LUname *llllll*

specifies the logical unit name of the corresponding real device for emulated 3270 display unit(s) or the logical unit name associated with the destination address of the emulated token-ring adapter.

DSname *nnnn...n[(mmm...m)]*

specifies the name of a sequential data set or the member name and name of a partitioned data set to be used to receive data for an emulated card reader device. If the operand is specified without data set name, then the device will be disassociated with the data set previously defined for the device. Specifies the name of a VSAM data set to be used to emulate a magnetic tape device. If the operand is specified without data set name, then the device will be disassociated with the data set previously defined for the device.

PROTECTED

defines that the emulated magnetic tape device has to be write protected. If the operand is specified, then the corresponding VSAM data set is allocated with DISP=SHR. If it is not specified, then it is allocated with DISP=OLD.

Class c

specifies the class of the SYSOUT data set to be used for emulation of a card punch or printer device. If the operand is specified without SYSOUT class, then the device will be disassociated with the SYSOUT class previously defined for the device.

LUNAME List

gives the full list of logical unit names assigned to emulated 3270 display units or associated with destination addresses of emulated token-ring adapters.

The following responses can be issued as the result of executing the TAKE LUNAME LIST command for emulated 3270 display units:

ADDRESS	LUNAME	STATE
aaaa	nnnnnnnn	c
...

Where 'c' is one of the following device state codes:

- A - device is active,
- R - device is ready for I/O,
- N - device is not operational.

The following responses can be issued as the result of executing the TAKE LUNAME LIST command for emulated token-ring adapters:

ADDRESS	LUNAME	STATE
aaaaaaaaaaaa	nnnnnnnn	c
...

Where 'c' is blank or one of the following station state codes:

- A - session is allocated.

ALLOCATION ccc/sap List

defines the allocation of *ccc* connections for service access point *sap* of the guest software for the emulated token-ring adapter with the guest device number *devaddr*. If connection allocation is not defined for a service access point, then the average number of currently available connections will be allocated. Connection allocation senses if the guest adapter is emulated by ISX via a real LSA adapter. The keyword 'List', if specified, gives the full list of the allocated connections for service access points of the guest software for the emulated token-ring adapter with the guest device number *devaddr*.

The following responses can be issued as the result of executing the TAKE xxxx ALLOCATION LIST command:

Connections	SAP	Status
ccc	sap	t
...

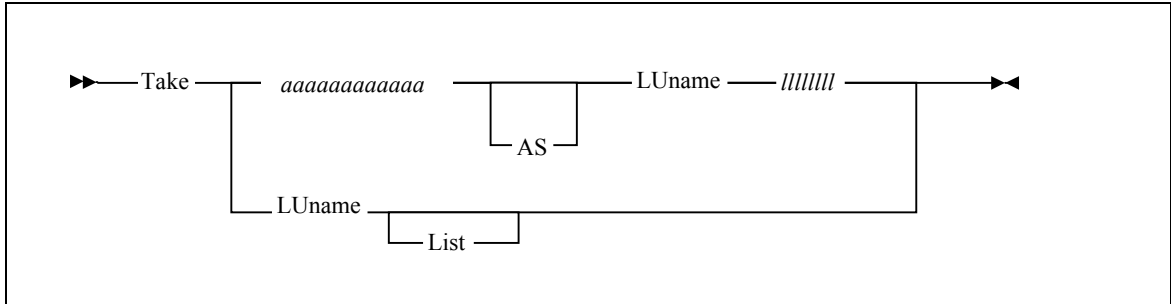
Where 't' is '-' or one of the following status codes:

- A - service access point sap is activated,
- D - service access point sap is deactivated.

IOBuffers nn

defines that nn I/O buffers have to be used for the emulated FBA DASD, CKD DASD, or magnetic tape. Maximal value for the number of I/O buffers is 99.

TAKE command format for ISX/LSA and ISX/LCS:



aaaaaaaaaaaa

defines destination hardware address of emulated token-ring adapter that has to be associated with the specified LU name.

Luname /////

specifies the logical unit name associated with the destination address of the emulated token-ring adapter.

Luname List

gives the full list of logical unit names associated with destination addresses of an emulated token-ring adapter.

The following responses can be issued as the result of executing the TAKE LUNAME LIST command:

ADDRESS	LUNAME	STATE
aaaaaaaaaaaa	nnnnnnnn	c
...

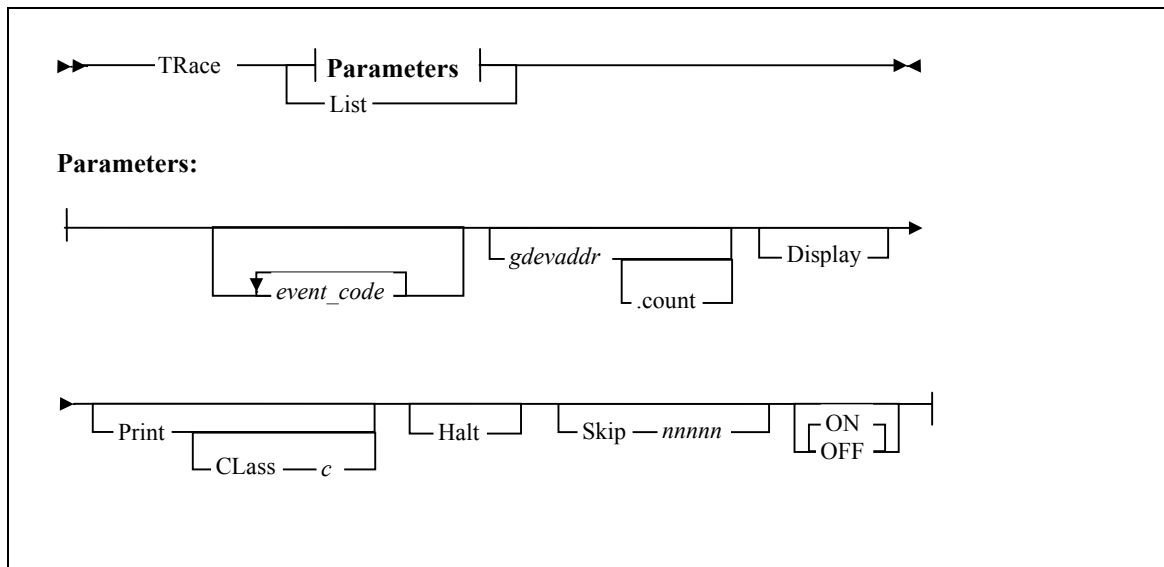
Where 'c' is blank or one of the following station state codes:

A - session is allocated.

TRACE Command

The TRACE command is designed to receive information about events happening in the guest system that are visible to ISX.

TRACE command format:



event_code

for S/390 architecture the following codes can be specified to define the guest events that are to be traced:

ITC	- unknown interception code
WAIT	- guest system in wait state
LCTL	- the LCTL instruction issued by the guest system
SIGP	- the SIGP instruction issued by the guest system
DIAG	- the DIAG instruction issued by the guest system
RRBE	- the RRBE instruction issued by the guest system
SSKE	- the SSKE instruction issued by the guest system
ISKE	- the ISKE instruction issued by the guest system
STIDP	- the STIDP instruction issued by the guest system
SCK	- the SCK instruction issued by the guest system
SPX	- the SPX instruction issued by the guest system
STPX	- the STPX instruction issued by the guest system
STAP	- the STAP instruction issued by the guest system
SERVC	- the SERVC instruction issued by the guest system
TB	- the TB instruction issued by the guest system
CSCH	- the CSCH instruction issued by the guest system
HSCH	- the HSCH instruction issued by the guest system
XSCH	- the XSCH instruction issued by the guest system
RSCH	- the RSCH instruction issued by the guest system
MSCH	- the MSCH instruction issued by the guest system
SAL	- the SAL instruction issued by the guest system
SSCH	- the SSCH instruction issued by the guest system
STSCH	- the STSCH instruction issued by the guest system
TSCH	- the TSCH instruction issued by the guest system
STCRW	- the STCRW instruction issued by the guest system
STCPS	- the STCPS instruction issued by the guest system
RCHP	- the RCHP instruction issued by the guest system

SCHM	- the SCHM instruction issued by the guest system
TPI	- the TPI instruction issued by the guest system
I/O	- I/O interruption occurred
EXT	- external interruption occurred
M/C	- machine-check interruption occurred
P/C	- program-check interruption occurred
CCW	- channel program executed by the guest system
STSI	- the STSI instruction issued by the guest system
SCKPF	- the SCKPF instruction issued by the guest system

For S/370 architecture the following codes can be specified to define the guest events that are to be traced:

ITC	- unknown interception code
LPSW	- the LPSW instruction issued by the guest system
PC	- the PC instruction issued by the guest system
PT	- the PT instruction issued by the guest system
SSM	- the SSM instruction issued by the guest system
STCTL	- the STCTL instruction issued by the guest system
LCTL	- the LCTL instruction issued by the guest system
DIAG	- the DIAG instruction issued by the guest system
STIDP	- the STIDP instruction issued by the guest system
SCK	- the SCK instruction issued by the guest system
SCKC	- the SCKC instruction issued by the guest system
STCKC	- the STCKC instruction issued by the guest system
SPT	- the SPT instruction issued by the guest system
STPT	- the STPT instruction issued by the guest system
SPKA	- the SPKA instruction issued by the guest system
IPK	- the IPK instruction issued by the guest system
PTLB	- the PTLB instruction issued by the guest system
RRB	- the RRB instruction issued by the guest system
RRBE	- the RRBE instruction issued by the guest system
SSK	- the SSK instruction issued by the guest system
SSKE	- the SSKE instruction issued by the guest system
ISK	- the ISK instruction issued by the guest system
ISKE	- the ISKE instruction issued by the guest system
IVSK	- the IVSK instruction issued by the guest system
IPTE	- the IPTE instruction issued by the guest system
SAC	- the SAC instruction issued by the guest system
IAC	- the IAC instruction issued by the guest system
SSAR	- the SSAR instruction issued by the guest system
EPAR	- the EPAR instruction issued by the guest system
ESAR	- the ESAR instruction issued by the guest system
FRR	- the FRR instruction issued by the guest system
MVCK	- the MVCK instruction issued by the guest system
MVCP	- the MVCP instruction issued by the guest system
MVCS	- the MVCS instruction issued by the guest system
LASP	- the LASP instruction issued by the guest system
TPROT	- the TPROT instruction issued by the guest system
STOSM	- the STOSM instruction issued by the guest system

STNSM	- the STNSM instruction issued by the guest system
TB	- the TB instruction issued by the guest system
SERV	- the SERVC instruction issued by the guest system
XSCH	- the XSCH instruction issued by the guest system
PRIV	- privileged instruction issued by the guest system
SVC	- the SVC instruction issued by the guest system
CLRIO	- the CLRIO instruction issued by the guest system
TIO	- the TIO instruction issued by the guest system
HDV	- the HDV instruction issued by the guest system
HIO	- the HIO instruction issued by the guest system
CLRCH	- the CLRCH instruction issued by the guest system
TCH	- the TCH instruction issued by the guest system
SIO	- the SIO instruction issued by the guest system
SIOF	- the SIOF instruction issued by the guest system
STIDC	- the STIDC instruction issued by the guest system
I/O	- I/O interruption occurred
EXT	- external interruption occurred
P/C	- program-check interruption occurred
CCW	- channel program executed by the guest system

If no event code(s) are specified, then all are assumed.

The following *gdevaddr* and *.count* operands can be specified to limit tracing to specific guest I/O devices:

gdevaddr

defines that I/O instructions and interruptions have to be traced for the guest device with the device number of *gdevaddr*.

.count

is a decimal value designating the number of devices starting with the device number of *gdevaddr* for which I/O instructions and interruptions have to be traced.

Display

defines that an event has to be displayed on the ISX operator console.

Print

defines that an event has to be printed to a SYSOUT data set. The SYSOUT data set is dynamically allocated by the TRACE command. When 20,000 records are written to the data set it is automatically closed and unallocated, and a new SYSOUT data set is allocated, opened, and used to contain subsequent TRACE records.

Class *c*

defines that a SYSOUT data set has class of *c*. The default value for the SYSOUT class is T.

Halt

defines that the guest system has to be stopped when an event has occurred.

Skip *nnnnn*

defines that the trace functions requested have to be activated, when *nnnnn* traced events have occurred.

ON
 defines that previously specified trace parameters are set on.

OFF
 defines that previously specified trace parameters are set off.

If neither ON nor OFF is specified, then ON assumed. New TRACE operands are logically added to or subtracted from the active operand combination. A TRACE command that specifies only the operand OFF indicates that the trace function is to be stopped.

List
 gives the full list of parameters currently defined for the trace function.

The ISX025I messages can be displayed and/or printed as the result of execution the TRACE command.

XBEGIN Command

The XBEGIN command is designed to continue the execution of ISX, ISX/LSA, or ISX/LCS when it is stopped by the request of the XSTOP command.

XBEGIN command format:

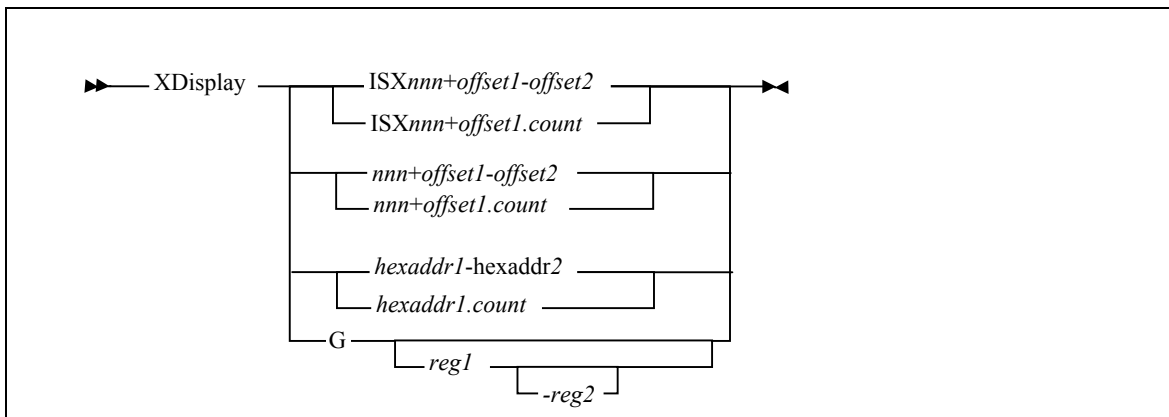


XDISPLAY Command

The XDISPLAY command is designed to display the following data from the ISX, ISX/LSA, or ISX/LCS environment:

- ISX, ISX/LSA, ISX/LCS virtual storage contents.
- ISX, ISX/LSA, ISX/LCS general registers contents.

XDISPLAY command format:



ISXnnn

defines full name of program module has to be displayed.

nnn

defines three last characters of program module has to be displayed.

+offset1

is the hexadecimal location offset in the specified program module that is to be displayed. The specified storage contents are displayed in hexadecimal and with EBCDIC translation.

-offset2

is the last of the range of hexadecimal location offset, whose contents are to be displayed.

hexaddr1

is the hexadecimal location address that is to be displayed. The specified storage contents are displayed in hexadecimal and with EBCDIC translation.

-hexaddr2

is the last of the range of hexadecimal location addresses, whose contents are to be displayed.

.count

is a hexadecimal value designating the number of bytes of storage, whose contents are to be displayed.

G[reg1[-reg2]]

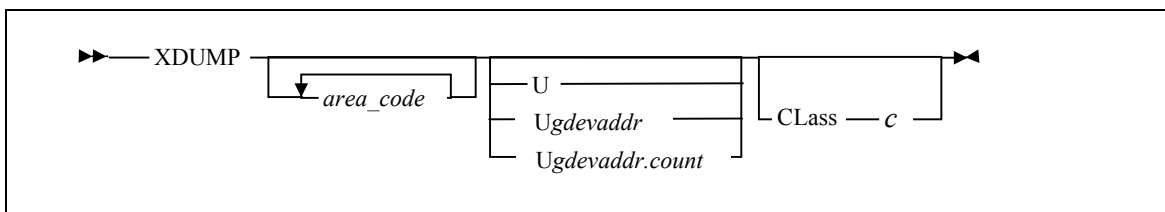
displays the contents of the general registers. If G is specified without a register number, the contents of all general registers are to be displayed. If only reg1 is specified, then only reg1 contents are to be displayed. If both reg1 and reg2 are specified, then the contents of all general registers from reg1 to reg2 are to be displayed.

XDUMP Command

The XDUMP command is designed to produce an ISX, ISX/LSA, or ISX/LCS storage dump to a dynamically allocated SYSOUT data set. The dump includes part of or all data for the following areas:

- Main ISX , ISX/LSA, or ISX/LCS control area,
- Common area,
- ISX , ISX/LSA, or ISX/LCS program area,
- Guest device control area(s).

XDUMP command format:



The XDUMP command without operands creates full storage dump.

area_code

the following codes can be specified to define storage areas:

ISCA - Main ISX control area,

DSCA - Main ISX/LSA control area,

ASCA - Main ISX/LCS control area,

COMM - Common area,

ISX - ISX program area.

LSA - ISX/LSA program area.

LCS - ISX/LCS program area.

U

guest device control areas for all guest I/O devices.

Ugdevaddr

guest device control area for guest I/O device with the device number of *gdevaddr*.

Ugdevaddr.count

guest device control areas for guest I/O devices beginning from the guest device with the device number of *gdevaddr*. '*count*' is a decimal value designating the number of devices starting with the device number of *gdevaddr* for which guest device control areas have to be included in the dump.

Class *c*

defines the SYSOUT class of data sets associated with the dump. The default value is D.

XSTOP Command

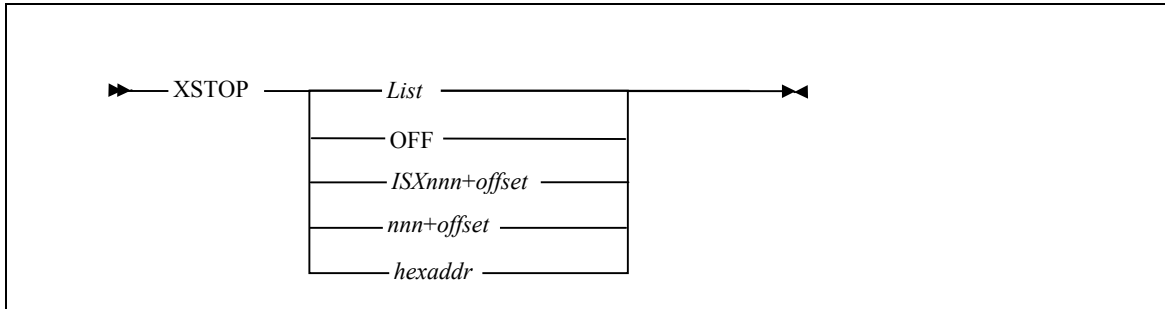
The XSTOP command is designed to stop ISX, ISX/LSA, or ISX/LCS execution at specified point. Only one stop point can be defined at a time. When ISX is stopped the following ISX commands can be issued from the ISX operator console:

- XBEGIN,
- XDISPLAY,
- XDUMP,
- XSTOP,
- XSTORE,
- BEGIN,
- DISPLAY,
- DUMP,
- SHUTDOWN,
- STOP,
- STORE,
- TRACE.

When ISX/LSA or ISX/LCS is stopped the following commands can be issued from the operator console:

- XBEGIN,
- XDISPLAY,
- XDUMP,
- XSTOP,
- XSTORE,
- BEGIN,
- SHUTDOWN.

XSTOP command format:



List

gives the previously defined stop point.

OFF

defines that previously defined stop point has to be cancelled.

ISXnnn

defines full name of program module has to be stopped.

nnn

defines three last characters of program module has to be stopped.

+offset

defines offset of program module where ISX, ISX/LSA, ISX/LCS has to be stopped.

hexaddr

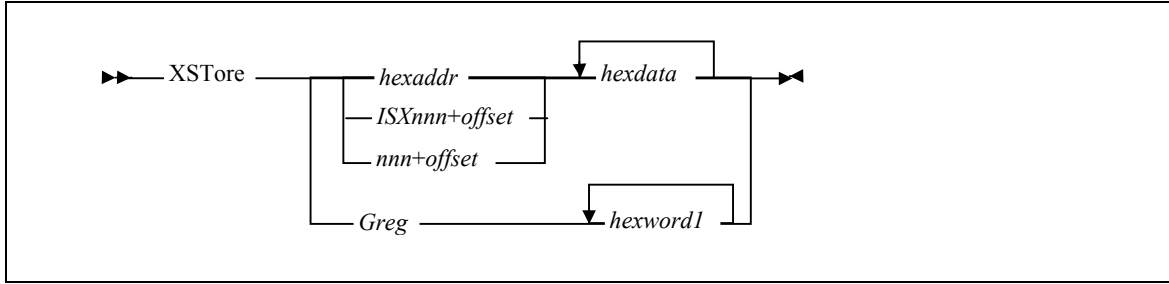
defines virtual address where ISX, ISX/LSA, ISX/LCS has to be stopped.

XSTORE Command

The XSTORE command is designed to alter the following ISX, ISX/LSA, or ISX/LCS environment:

- ISX, ISX/LSA, ISX/LCS virtual storage contents,
- ISX, ISX/LSA, ISX/LCS general register contents.

XSTORE command format:



ISXnnn

defines full name of program module that is to be altered.

nnn

defines three last characters of program module that is to be altered.

+offset

is the hexadecimal location offset in the specified program module that is to be altered.

hexaddr

is the hexadecimal location address that is to be altered.

hexdata

stores the hexadecimal data (*hexdata*) starting at the address specified by the previous operand.

Greg hexword1 ...

stores the hexadecimal word(s) of data in successive general registers starting at the register specified by reg.

MESSAGES

ISX, ISX/LSA, and ISX/LCS messages have the following message format:

ISXnnns text

Where:

ISX - ISX program product name,

nnn - message serial number,

s - message type code:

I - information; no operator action is required,

A - action; the operator must perform a specific action,

W - warning; action has to be determined and performed.

text - message text.

ISX001I Guest is running

This message is issued when the guest system is executed by ISX.

ISX002W Device xxxx - I/O buffer overflow. Channel program is cut down

This message is issued when ISX detects that translation buffers used to convert a guest channel program for xxxx device to zOS form has overflowed. The Guest channel program is truncated.

ISX003W Password and expiration date must be defined before IPL

This message is issued when initial program loading of the guest system is requested before definition of the customer's password and expiration date for ISX.

ISX004I Trace file nnn for jjjjjjjj is unallocated

This message is issued when current trace file nnn of ISX job jjjjjjjj is closed and unallocated.

ISX005I Guest is stopped [... at address xxxxxx]

This message is issued when ISX stops execution of the guest system by request of ISX operator.

ISX006I Initial reset has done

This message is issued when during execution of the IPL command with the NOCLEAR operand store CPU status and initial CPU reset operations have been performed.

ISX007I File nnn for xxxx, jjjjjjjj is unallocated

This message is issued when the current log data set nnn of the emulated token-ring adapter of ISX, ISX/LSA, or ISX/LCS job jjjjjjjj is closed and unallocated. The value xxxx is the guest device number for ISX or is the real channel-to-channel adapter device number for ISX/LSA or ISX/LCS.

ISX008I Device xxxx is ready for I/O

This message is issued when ISX detects that the specified device xxxx is operational.

ISX009I Device xxxx is active

This message is issued when ISX detects that the specified device xxxx is doing I/O. If the message is issued during execution of the SAVESYS or TAKE command, then reissue the corresponding command at a later time.

ISX010W Device xxxx is not ready for I/O

This message is issued for a real device with the guest device number xxxx, when the READY Command Processor detects that the device is not operational.

This message is issued for an emulated device, when the guest system or ISX has issued an I/O request, but ISX can not allocate or open the data set defined with the DEFINE command for the device or receive storage for the input buffer.

ISX011W ATTENTION! VALIDITY OF THE ISX/pgm LICENCE HAS ELAPSED

This message is issued when the ISX license has expired.

ISX012W VALIDITY OF THE ISX/pgm LICENSE IS EXPIRING

This message is issued 30 days prior to the ISX license expiration date.

ISX013A Console nnnnnnnn is not active

This message is issued during execution of the SET command if the specified console nnnnnnnn is not active.

The ISX operator action: Activate the zOS console nnnnnnnn and reissue the SET command or specify another console name in the SET command.

ISX014A Console validation service not available

This message is issued during execution of the SET command if the console validation service of the CONVCON macro is not available.

The ISX operator action: Resubmit the request at a later time.

ISX015A Specified console name nnnnnnnn not defined in zOS

This message is issued during execution of the SET command if the specified console name nnnnnnnn is not defined in zOS.

The ISX operator action: Specify a console name is defined in zOS.

ISX016W CONVCON failed

This message is issued during execution of the SET command if any error is detected during execution of the CONVCON macro.

ISX017W DYNALLOC failed. RC=rr EC=eeee IC=iiii

This message is issued if any error is detected during execution of the DYNALLOC macro. RC is defined the return code, EC is defined the error code, IC is defined the information code.

ISX018W Incorrect state of ISX Common Area. Reply R to reinitialize it or
U to use it as is.

This message is issued to the ISX operator if an incorrect state of the ISX Common Area is detected during ISX, ISX/LSA, or ISX/LCS initialization. The reply 'R' is possible if ISX is alone in the system. The reply 'U' is possible, if another ISX is running.

ISX019A Invalid reply, try again.

This message is issued to the ISX operator if an incorrect reply is entered in response to the ISX018W message.

The ISX operator action: Enter correct reply to the request.

ISX020W ISX/cmp, jjjjjjjj abnormally terminated

This message is issued when the ISX/cmp component of jjjjjjjj job is abnormally terminated. The cmp value defines one of the following components of ISX:

- ISX/370 - ISX itself;
- ISX/390 - ISX itself;
- ISX/LSA - The LSA Emulator;
- ISX/LCS - The LCS Emulator;
- ISX/URD - The Unit Record Device Emulator;
- ISX/FBA - The FBA DASD Emulator;
- ISX/CKD - The CKD DASD Emulator;
- ISX/MTP - The Magnetic Tape Emulator;
- ISX/GRF - The Display Unit Emulator;
- ISX/CET - The CET Token-Ring Adapter Emulator;
- ISX/LCS - The LCS Token-Ring Adapter Emulator;
- ISX/CTC - The Channel-To-Channel Adapter Emulator.

ISX021W No SDWA has built by zOS

This message is issued to the ISX operator if no SDWA was built for the ESTAE Exit routine of ISX.

ISX022W Completion code ccccc Reason code rrrrrrr

This message is issued to the ISX operator to display a completion code ccccc and a reason code rrrrrrr (if the last one is supported).

ISX023W Percolation has occurred

This message is issued to the ISX operator if percolation has occurred, when the ESTAE Exit routine of ISX receives control from zOS.

ISX024W Unexpected interception with code cc

This message is issued when ISX receives interception with an interception code of cc. The guest system has stopped.

ISX025I message text

These messages are displayed or printed when the ISX trace function is active with DISPLAY or PRINT option. The 'message text' is one of the following:

- ITC cc PSW xxxxxxxx xxxxxxxx
 - defines that interception of the guest system with the invalid code cc has occurred.
 The guest PSW is xxxxxxxx xxxxxxxx.
- WAIT mm PSW xxxxxxxx xxxxxxxx
 - defines that the guest system is in a wait state.
 The guest I/O interruption subclass mask is mm.
 The guest PSW is xxxxxxxx xxxxxxxx.
- LPSW NPSW yyyyyyyy yyyyyyyy PSW xxxxxxxx xxxxxxxx
 - defines that the LPSW instruction is issued by the guest system.
 The guest new PSW is yyyyyyyy yyyyyyyy.
 The guest PSW is xxxxxxxx xxxxxxxx.
- PC NPSW yyyyyyyy yyyyyyyy PSW xxxxxxxx xxxxxxxx
 - defines that the PC instruction is issued by the guest system.
 The guest new PSW is yyyyyyyy yyyyyyyy.
 The guest PSW is xxxxxxxx xxxxxxxx.
- PT NPSW yyyyyyyy yyyyyyyy PSW xxxxxxxx xxxxxxxx
 - defines that the PT instruction is issued by the guest system.
 The guest new PSW is yyyyyyyy yyyyyyyy.
 The guest PSW is xxxxxxxx xxxxxxxx.
- SSM PSW xxxxxxxx xxxxxxxx
 - defines that the SSM instruction is issued by the guest system.
 The guest PSW is xxxxxxxx xxxxxxxx.
- LCTL x,y aaaaaaa PSW xxxxxxxx xxxxxxxx
 - defines that the LCTL instruction is issued by the guest system.
 Control registers from x to y are the instruction operands.
 The control registers contents are at aaaaaaa location.
 The guest PSW is xxxxxxxx xxxxxxxx.
- STCTL x,y aaaaaaa PSW xxxxxxxx xxxxxxxx
 - defines that the STCTL instruction is issued by the guest system.
 Control registers from x to y are the instruction operands.
 The control registers contents are stored at aaaaaaa location.
 The guest PSW is xxxxxxxx xxxxxxxx.
- SIGP x,y aaaaaaa PSW xxxxxxxx xxxxxxxx
 - defines that the SIGP instruction is issued by the guest system.
 The instruction operands are x, y, aaaaaaa.
 The guest PSW is xxxxxxxx xxxxxxxx.
- DIAG x,y aaaaaaa PSW xxxxxxxx xxxxxxxx
 - defines that the DIAG instruction is issued by the guest system.
 The instruction operands are x, y, aaaaaaa.
 The guest PSW is xxxxxxxx xxxxxxxx.
- STIDP aaaaaaa PSW xxxxxxxx xxxxxxxx
 - defines that the STIDP instruction is issued by the guest system.
 The instruction operand is aaaaaaa.
 The guest PSW is xxxxxxxx xxxxxxxx.

72 MESSAGES

SCK aaaaaaaaa PSW xxxxxxxx xxxxxxxx

- defines that the SCK instruction is issued by the guest system.
The instruction operand is aaaaaaaaa.
The guest PSW is xxxxxxxx xxxxxxxx.

SCKC aaaaaaaaa PSW xxxxxxxx xxxxxxxx

- defines that the SCKC instruction is issued by the guest system.
The instruction operand is aaaaaaaaa.
The guest PSW is xxxxxxxx xxxxxxxx.

SCKPF vvvv PSW xxxxxxxx xxxxxxxx

- defines that the SCKPF instruction is issued by the guest system.
The instruction operand is vvvv.
The guest PSW is xxxxxxxx xxxxxxxx.

STCKC aaaaaaaaa PSW xxxxxxxx xxxxxxxx

- defines that the STCKC instruction is issued by the guest system.
The instruction operand is aaaaaaaaa.
The guest PSW is xxxxxxxx xxxxxxxx.

SPT aaaaaaaaa PSW xxxxxxxx xxxxxxxx

- defines that the SPT instruction is issued by the guest system.
The instruction operand is aaaaaaaaa.
The guest PSW is xxxxxxxx xxxxxxxx.

STPT aaaaaaaaa PSW xxxxxxxx xxxxxxxx

- defines that the STPT instruction is issued by the guest system.
The instruction operand is aaaaaaaaa.
The guest PSW is xxxxxxxx xxxxxxxx.

SPKA aaaaaaaaa PSW xxxxxxxx xxxxxxxx

- defines that the SPKA instruction is issued by the guest system.
The instruction operand is aaaaaaaaa.
The guest PSW is xxxxxxxx xxxxxxxx.

IPK PSW xxxxxxxx xxxxxxxx

- defines that the IPK instruction is issued by the guest system.
The guest PSW is xxxxxxxx xxxxxxxx.

PTLB PSW xxxxxxxx xxxxxxxx

- defines that the PTLB instruction is issued by the guest system.
The guest PSW is xxxxxxxx xxxxxxxx.

RRB aaaaaaaaa PSW xxxxxxxx xxxxxxxx CC z

- defines that the RRB instruction is issued by the guest system.
The instruction operand is aaaaaaaaa.
The guest PSW is xxxxxxxx xxxxxxxx.
Condition code is z.

RRBE aaaaaaaaa PSW xxxxxxxx xxxxxxxx CC z

- defines that the RRBE instruction is issued by the guest system.
The instruction operand is aaaaaaaaa.
The guest PSW is xxxxxxxx xxxxxxxx.
Condition code is z.

- SSK kk aaaaaaaa PSW xxxxxxxx xxxxxxxx
 - defines that the SSK instruction is issued by the guest system.
 Storage protection key is kk.
 The instruction operand is aaaaaaaa.
 The guest PSW is xxxxxxxx xxxxxxxx.
- SSKE kk aaaaaaaa PSW xxxxxxxx xxxxxxxx
 - defines that the SSKE instruction is issued by the guest system.
 Storage protection key is kk.
 The instruction operand is aaaaaaaa.
 The guest PSW is xxxxxxxx xxxxxxxx.
- ISK kk aaaaaaaa PSW xxxxxxxx xxxxxxxx
 - defines that the ISK instruction is issued by the guest system.
 Storage protection key is kk.
 The instruction operand is aaaaaaaa.
 The guest PSW is xxxxxxxx xxxxxxxx.
- ISKE kk aaaaaaaa PSW xxxxxxxx xxxxxxxx
 - defines that the ISKE instruction is issued by the guest system.
 Storage protection key is kk.
 The instruction operand is aaaaaaaa.
 The guest PSW is xxxxxxxx xxxxxxxx.
- IVSK kk aaaaaaaa PSW xxxxxxxx xxxxxxxx
 - defines that the IVSK instruction is issued by the guest system.
 Storage protection key is kk.
 The instruction operand is aaaaaaaa.
 The guest PSW is xxxxxxxx xxxxxxxx.
- IPTE aaaaaaa PSW xxxxxxxx xxxxxxxx
 - defines that the IPTE instruction is issued by the guest system.
 The guest page table element address is aaaaaaaa.
 The guest PSW is xxxxxxxx xxxxxxxx.
- SAC PSW xxxxxxxx xxxxxxxx
 - defines that the SAC instruction is issued by the guest system.
 The guest PSW is xxxxxxxx xxxxxxxx.
- IAC PSW xxxxxxxx xxxxxxxx CC z
 - defines that the IAC instruction is issued by the guest system.
 The guest PSW is xxxxxxxx xxxxxxxx.
 Condition code is z.
- SSAR ASN nnnn PSW xxxxxxxx xxxxxxxx
 - defines that the SSAR instruction is issued by the guest system.
 New secondary ASN is nnnn.
 The guest PSW is xxxxxxxx xxxxxxxx.
- EPAR ASN nnnn PSW xxxxxxxx xxxxxxxx
 - defines that the EPAR instruction is issued by the guest system.
 Current primary ASN is nnnn.
 The guest PSW is xxxxxxxx xxxxxxxx.

74 MESSAGES

ESAR ASN nnnn PSW xxxxxxxx xxxxxxxx

- defines that the ESAR instruction is issued by the guest system.
Current secondary ASN is nnnn.
The guest PSW is xxxxxxxx xxxxxxxx.

FRR PSW xxxxxxxx xxxxxxxx

- defines that the FRR instruction is issued by the guest system.
The guest PSW is xxxxxxxx xxxxxxxx.

MVCK PSW xxxxxxxx xxxxxxxx CC z

- defines that the MVCK instruction is issued by the guest system.
The guest PSW is xxxxxxxx xxxxxxxx.
Condition code is z.

MVCP PSW xxxxxxxx xxxxxxxx CC z

- defines that the MVCP instruction is issued by the guest system.
The guest PSW is xxxxxxxx xxxxxxxx.
Condition code is z.

MVCS PSW xxxxxxxx xxxxxxxx CC z

- defines that the MVCS instruction is issued by the guest system.
The guest PSW is xxxxxxxx xxxxxxxx.
Condition code is z.

LASP PSW xxxxxxxx xxxxxxxx CC z

- defines that the LASP instruction is issued by the guest system.
The guest PSW is xxxxxxxx xxxxxxxx.
Condition code is z.

TPROT PSW xxxxxxxx xxxxxxxx CC z

- defines that the TPROT instruction is issued by the guest system.
The guest PSW is xxxxxxxx xxxxxxxx.
Condition code is z.

STOSM PSW xxxxxxxx xxxxxxxx

- defines that the STOSM instruction is issued by the guest system.
The guest PSW is xxxxxxxx xxxxxxxx.

STNSM PSW xxxxxxxx xxxxxxxx

- defines that the STNSM instruction is issued by the guest system.
The guest PSW is xxxxxxxx xxxxxxxx.

STSI aaaaaaa PSW xxxxxxxx xxxxxxxx

- defines that the STSI instruction is issued by the guest system.
The instruction operand is aaaaaaa.
The guest PSW is xxxxxxxx xxxxxxxx.

SPX pppppppp PSW xxxxxxxx xxxxxxxx

- defines that the SPX instruction is issued by the guest system.
The prefix value is pppppppp.
The guest PSW is xxxxxxxx xxxxxxxx.

- STPX pppppppp PSW xxxxxxxx xxxxxxxx
 - defines that the STPX instruction is issued by the guest system.
 The prefix value is pppppppp.
 The guest PSW is xxxxxxxx xxxxxxxx.
- STAP aaaaaaaaa PSW xxxxxxxx xxxxxxxx
 - defines that the STAP instruction is issued by the guest system.
 The instruction operand is aaaaaaaaa.
 The guest PSW is xxxxxxxx xxxxxxxx.
- SERVC x,y PSW xxxxxxxx xxxxxxxx
 - defines that the SERVC instruction is issued by the guest system.
 The instruction operands are x, y.
 The guest PSW is xxxxxxxx xxxxxxxx.
- TB aaaaaaaaa PSW xxxxxxxx xxxxxxxx CC z
 - defines that the TB instruction is issued by the guest system.
 The instruction operand is aaaaaaaaa.
 The guest PSW is xxxxxxxx xxxxxxxx.
 Condition code is z.
- SVC nn PSW xxxxxxxx xxxxxxxx
 - defines that the SVC instruction is issued by the guest system.
 SVC number is nn.
 The guest PSW is xxxxxxxx xxxxxxxx.
- CSCH dddd PSW xxxxxxxx xxxxxxxx CC z
 - defines that the CSCH instruction is issued by the guest system.
 The device number is dddd.
 The guest PSW is xxxxxxxx xxxxxxxx.
 Condition code is z.
- HSCH dddd PSW xxxxxxxx xxxxxxxx CC z
 - defines that the HSCH instruction is issued by the guest system.
 The device number is dddd.
 The guest PSW is xxxxxxxx xxxxxxxx.
 Condition code is z.
- XSCH dddd PSW xxxxxxxx xxxxxxxx CC z
 - defines that the XSCH instruction is issued by the guest system.
 The device number is dddd.
 The guest PSW is xxxxxxxx xxxxxxxx.
 Condition code is z.
- RSCH dddd PSW xxxxxxxx xxxxxxxx CC z
 - defines that the RSCH instruction is issued by the guest system.
 The device number is dddd.
 The guest PSW is xxxxxxxx xxxxxxxx.
 Condition code is z.

- MSCH dddd aaaaaaaa PSW xxxxxxxx xxxxxxxx CC z
- defines that the MSCH instruction is issued by the guest system.
 - The device number is dddd.
 - The guest subchannel information block address is aaaaaaaa.
 - The guest PSW is xxxxxxxx xxxxxxxx.
 - Condition code is z.
- ... mmff pppppppp
- defines the continuation of the MSCH instruction record, if the instruction has performed with CC=0.
 - The interruption subclass code is mm.
 - The path management control word flags are ff.
 - The I/O interruption parameter is pppppppp.
- SAL llllllll PSW xxxxxxxx xxxxxxxx
- defines that the SAL instruction is issued by the guest system.
 - The guest address limit value is llllllll.
 - The guest PSW is xxxxxxxx xxxxxxxx.
- SSCH dddd aaaaaaaa PSW xxxxxxxx xxxxxxxx CC z
- defines that the SSCH instruction is issued by the guest system.
 - The device number is dddd.
 - The guest operation request block address is aaaaaaaa.
 - The guest PSW is xxxxxxxx xxxxxxxx.
 - Condition code is z.
- ... ORB xxxxxxxx xxxxxxxx xxxxxxxx
- defines the continuation of the SSCH instruction record, if the instruction has performed with CC=0.
 - The operation request block is xxxxxxxx xxxxxxxx xxxxxxxx.
- dddd CCW aaaaaaaa xxxxxxxx xxxxxxxx
- defines that the guest channel program has the CCW.
 - The device number is dddd.
 - The guest address of the CCW is aaaaaaaa.
 - The guest CCW is xxxxxxxx xxxxxxxx.
- SENSE DATA nn xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx[...]
- defines that guest I/O device has the sense data.
 - The number of sense data bytes is nn.
 - The sense data bytes is xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx.
- dddd IDAW aaaaaaaa xxxxxxxx xxxxxxxx
- defines that the guest channel program has the IDAW.
 - The device number is dddd.
 - The guest address of the IDAW is aaaaaaaa.
 - The guest IDAW is xxxxxxxx xxxxxxxx.
- STSCH dddd aaaaaaaa PSW xxxxxxxx xxxxxxxx CC z
- defines that the STSCH instruction is issued by the guest system.
 - The device number is dddd.
 - The guest subchannel information block address is aaaaaaaa.
 - The guest PSW is xxxxxxxx xxxxxxxx.
 - Condition code is z.

- ... SCSW xxxxxxxx xxxxxxxx xxxxxxxx
- defines the continuation of the STSCH instruction record, if the instruction has performed with CC=0 and data stored in the subchannel information block.
The subchannel status word is xxxxxxxx xxxxxxxx xxxxxxxx.
- TSCH dddd aaaaaaaa PSW xxxxxxxx xxxxxxxx CC z
- defines that the TSCH instruction is issued by the guest system.
The device number is dddd.
The guest interruption response block address is aaaaaaaa.
The guest PSW is xxxxxxxx xxxxxxxx.
Condition code is z.
- ... SCSW xxxxxxxx xxxxxxxx xxxxxxxx
- defines the continuation of the TSCH instruction record, if the instruction has performed with CC=0 or CC=1 and none zero data stored in the interruption response block.
The subchannel status word is xxxxxxxx xxxxxxxx xxxxxxxx.
- STCRW aaaaaaaa PSW xxxxxxxx xxxxxxxx CC z
- defines that the STCRW instruction is issued by the guest system.
The guest location address for the channel report word is aaaaaaaa.
The guest PSW is xxxxxxxx xxxxxxxx.
Condition code is z.
- STCPS aaaaaaaa PSW xxxxxxxx xxxxxxxx
- defines that the STCPS instruction is issued by the guest system.
The guest location address for the channel path status is aaaaaaaa.
The guest PSW is xxxxxxxx xxxxxxxx.
- RCHP id PSW xxxxxxxx xxxxxxxx CC z
- defines that the RCHP instruction is issued by the guest system.
The channel path identifier is id.
The guest PSW is xxxxxxxx xxxxxxxx.
Condition code is z. SCHM bbbbbbbb PSW xxxxxxxx xxxxxxxx
- SCHM bbbbbbbb PSW xxxxxxxx xxxxxxxx
- defines that the SCHM instruction is issued by the guest system.
The measurement mode control bits are bbbbbbbb.
The guest PSW is xxxxxxxx xxxxxxxx.
- TPI mm aaaaaaaa PSW xxxxxxxx xxxxxxxx CC z
- defines that the TPI instruction is issued by the guest system.
The guest I/O interruption subclass mask is mm.
The guest I/O interruption code address is aaaaaaaa.
The guest PSW is xxxxxxxx xxxxxxxx.
Condition code is z.
- ... dddd pppppppp
- defines the continuation of the TPI instruction record, if the instruction has performed with CC=1.
The device number is dddd.
The I/O interruption parameter is pppppppp.
- CLRIO ddd CSW ccccccc ccccccc PSW xxxxxxxx xxxxxxxx CC z
- defines that the CLRIO instruction is issued by the guest system.
The device address is ddd.
The guest CSW is ccccccc ccccccc.

78 MESSAGES

The guest PSW is xxxxxxxx xxxxxxxx.
Condition code is z.

TIO ddd CSW ccccccc ccccccc PSW xxxxxxxx xxxxxxxx CC z

- defines that the TIO instruction is issued by the guest system.
The device address is ddd.
The guest CSW is ccccccc ccccccc.
The guest PSW is xxxxxxxx xxxxxxxx.
Condition code is z.

HIO ddd PSW xxxxxxxx xxxxxxxx CC z

- defines that the HIO or HDV instruction is issued by the guest system.
The device address is ddd.
The guest PSW is xxxxxxxx xxxxxxxx.
Condition code is z.

CLRCH c00 PSW xxxxxxxx xxxxxxxx CC z

- defines that the CLRCH instruction is issued by the guest system.
The channel address is c00.
The guest PSW is xxxxxxxx xxxxxxxx.
Condition code is z.

TCH c00 PSW xxxxxxxx xxxxxxxx CC z

- defines that the TCH instruction is issued by the guest system.
The channel address is c00.
The guest PSW is xxxxxxxx xxxxxxxx.
Condition code is z.

SIO ddd CAW aaaaaaa PSW xxxxxxxx xxxxxxxx CC z

- defines that the SIO instruction is issued by the guest system.
The device address is ddd.
The guest CAW is aaaaaaa.
The guest PSW is xxxxxxxx xxxxxxxx.
Condition code is z.

SIOF ddd CAW aaaaaaa PSW xxxxxxxx xxxxxxxx CC z

- defines that the SIOF instruction is issued by the guest system.
The device address is ddd.
The guest CAW is aaaaaaa.
The guest PSW is xxxxxxxx xxxxxxxx.
Condition code is z.

STIDC c00 CID iiiiii PSW xxxxxxxx xxxxxxxx CC z

- defines that the STIDC instruction is issued by the guest system.
The channel address is c00.
The channel ID is iiiiii.
The guest PSW is xxxxxxxx xxxxxxxx.
Condition code is z.

I/O dddd ppppppp PSW xxxxxxxx xxxxxxxx

- defines that I/O interruption is reflected to the guest system.
The device number is dddd.
The I/O interruption parameter is ppppppp.
The guest PSW is xxxxxxxx xxxxxxxx.

EXT cccc pppppppp PSW xxxxxxxx xxxxxxxx

- defines that external interruption is reflected to the guest system.
The interruption code is cccc.
The interruption parameter is pppppppp, if it is an service call external interruption.
The guest PSW is xxxxxxxx xxxxxxxx.

M/C PSW xxxxxxxx xxxxxxxx

- defines that machine-check interruption is reflected to the guest system.
The guest PSW is xxxxxxxx xxxxxxxx.

P/C cccc aaaaaaaa PSW xxxxxxxx xxxxxxxx

- defines that program interruption has occurred in the guest system.
The interruption code is cccc.
The guest page/segment fault address is aaaaaaaa.
The guest PSW is xxxxxxxx xxxxxxxx.

ISX025I ??? cccc PSW xxxxxxxx xxxxxxxx

- defines that operation exception program interruption has occurred in the guest system.
The operation code is cccc.
The guest PSW is xxxxxxxx xxxxxxxx.

ISX026I Program exception on instruction cccc IC=ii

This message is issued when ISX receives a program interruption on an instruction with operation code of cccc and interruption code of ii.

ISX027W ATTENTION! INCORRECT PASSWORD IS SPECIFIED FOR ISX/pgm

This message is issued when an incorrect password was specified for ISX.

ISX028W Device xxxx I/O error. op csw s

This message is issued when ISX detects an I/O error during execution of the guest I/O request for the device xxxx. In this message 'op' is operation code of the last executed CCW, 'csw' is the subchannel status word, 's' is the first sense byte.

ISX029W VTAM RPL-based macro error. LUNAME=llllllll, GRC=gc,
RTNCD=rc, FDB2=ec, SENSE=ssmmuuuu

This message is issued when VTAM returns control to the ISX with an error or special-condition information about the requested RPL-based operation. In this message 'LUNAME' defines the logical unit name of the device, 'GRC' defines the general return code, 'RTNCD' defines the recovery action return code, 'FDB2' defines the specific error return code or conditional completion return code, if 'RTNCD' defines zero. 'SENSE' is set only for certain RTNCD/FDB2 combinations; it contains system-sense information (the first byte), system-sense modifier information (the second byte) and user-sense information (the last two bytes).

ISX030A LU type of llllllll is not supported

This message is issued when an unsupported logical unit (neither LU type 0 nor LU type 2) is defined by the LUNAME operand of the DEFINE command.

The ISX operator action: Specify logical unit of type 0 or type 2 in the DEFINE command.

ISX031W Unpredictable IPL CCW sequence

This message is issued during execution of the IPL command if the first channel command word of the guest IPL channel program for an FBA disk is not READ IPL.

ISX032I Device xxxx detached

This message is issued when the device xxxx is detached.

ISX033I Device xxxx is reset

This message is issued when the device xxxx is reset.

ISX034I Clear reset has done

This message is issued when the guest environment is reset.

ISX035I Shutdown completed for ISX/cmp job jjjjjjjj

This message is issued when ISX, ISX/LSA, or ISX/LCS shutdown has completed.

ISX036A The command is not applicable to device xxxx

This message is issued when the ISX operator is trying to use the CLOSE, TAKE, or SET command for the device xxxx, but the command is not applicable to the device.

The ISX operator action: Check the ISX command syntax.

ISX037A Path to ddddddddddd allocated via LU name nnnnnnnn

This message is issued when the ISX operator is trying to redefine the LU name for partner ddddddddddd of a full emulated token-ring adapter, but ISX currently communicates with the partner via nnnnnnnn LU name. The request is rejected.

The ISX operator action: 1. Check partner's destination hardware address, specified in the TAKE command. 2. If you want to redefine the LU name for the destination hardware address ddddddddddd, then previously issue the TAKE command for this address without LU name to reject previous association of ddddddddddd with the LU name nnnnnnnn and, farther, reissue the command.

ISX038I Path to xxxx, jjjjjjjj from ddddddddddd 'text'

This message is issued when the data transfer path to the token-ring adapter xxxx of ISX job jjjjjjjj from the remote station with hardware address of aaaaaaaaaa is failed or rejected.

ISX039W Session llllllll terminated, device xxxx disconnected

This message is issued when the session with the VTAM terminal llllllll has been terminated due to I/O problems for an emulated 3270 display unit xxxx. The device xxxx remains operational for the guest system, but not is connected with a VTAM terminal.

ISX040A ISX/cmp job jjjjjjjj is not connected with VTAM

This message is issued if ISX, ISX/LSA, or ISX/LCS is not defined as the VTAM application program or I/O problems were detected while opening ISX's VTAM ACB or on the SETLOGON macro.

The ISX operator action: Check ISX application program is defined with an APPL definition statement, the name of the statement matches the job step name of the ISX startup job, and the corresponding VTAM node is active.

ISX041I Message from xxxx, jjjjjjjj >> mhmhmmh chchch pppp...pppp

This message is issued when the outbound message with the message header mhmhmmh, the CIOA header chchch, and the parameters pppp...pppp is sent by the guest system to ISX job jjjjjjjj via the guest outbound port xxxx of the emulated token-ring adapter.

ISX042I Message to xxxx, jjjjjjj << mhmhlmhm chchch pppp...pppp

This message is issued when the inbound message with the message header mhmhlmhm, the CIOA header chchch, and the parameters pppp...pppp is received by the guest system from ISX job jjjjjjj via the guest inbound port xxxx of the emulated token-ring adapter.

ISX043W Unknown message from xxxx, jjjjjjj

This message is issued when an unknown type message is received by ISX from the guest outbound port xxxx of ISX job jjjjjjj.

ISX044W Unknown frame to xxxx, jjjjjjj

This message is issued when an unknown type frame is received by ISX for the guest inbound port xxxx of ISX job jjjjjjj.

ISX045W Mismatch send/receive sequence numbers: ss/rr jjjjjjj

This message is issued when a mismatch ss / rr is detected between send and receive sequence numbers of data transfer process of ISX job jjjjjjj.

ISX046I Frame to xxxx, jjjjjjj is discarded

This message is issued when a frame is received by ISX for the guest inbound port xxxx of ISX job jjjjjjj, but was not reflected to the guest system as the message.

ISX047I Frame to xxxx, jjjjjjj << ddddddddddd ...

This message is issued when an inbound frame ddddddddddd ... is received by ISX job jjjjjjj from a real or virtual token-ring for the guest inbound port xxxx of the emulated token-ring adapter.

ISX048I Frame from xxxx, jjjjjjj >> ddddddddddd ...

This message is issued when an outbound frame ddddddddddd ... is sent by ISX job jjjjjjj to a real or virtual token-ring by request of the guest outbound port xxxx of the emulated token-ring adapter.

ISX049W IEANTCR failed

This message is issued during ISX initialization, if any errors are detected during execution of the IEANTCR macro.

ISX050W IEANTRT failed

This message is issued during ISX initialization, if any errors are detected during execution of the IEANTRT macro.

ISX051W Invalid frame to xxxx, jjjjjjj

This message is issued when an invalid frame is received by ISX for the guest inbound port xxxx of ISX job jjjjjjj.

ISX052A Path to ddddddddddd has no LU name in jjjjjjjj

This message is issued when the ISX operator is trying to reject the LU name for remote partner ddddddddddd or the guest system is trying to send data to remote partner ddddddddddd or ISX receives data from remote partner ddddddddddd, but the ISX job jjjjjjjj has no LU name associated with the partner.

The ISX operator action: Associate the destination hardware address ddddddddddd with the LU name of the appropriate partner with the TAKE command.

ISX053W zArchitecture hardware is required

This message is issued when ISX is started up not on zArchitecture computer.

ISX054W Adapter xxxx, port n initialization error

This message is issued if a negative response was received from the real token-ring adapter xxxx, port n during its initialization.

ISX056W VSAM mmmmmm macro failed. RC=x,EC=yyy,DSNAME=nnn...n

This message is issued if the VSAM macro named in this message has a non-zero return code. The VSAM macro return code is displayed in the RC field, the corresponding error code is displayed in the EC field, the corresponding data set name is displayed in the DSNAME field of this message.

ISX060A Save the volume in the guest system

This message is issued when I/O error is detected on write operation mentioned in the ISX056W message.

The guest system operator action: The corresponding guest volume must be saved in the guest system.

ISX061W Inconsistent VSAM data set attributes

This message is issued during execution of the SAVESYS command or the DEFINE command, if the specified data set is not a VSAM ESDS data set or the maximum record length of the specified data set is not correct.

ISX062A Enter VSAM data set name

This message is issued during VSAM data set loading procedure to prompt the operator to enter the name of the VSAM data set.

The ISX operator action: Enter a VSAM data set name.

ISX063A No data set allocated

This message is issued during VSAM data set loading if the VSAM data set specified cannot be allocated.

ISX064A Confirm request for loading. Reply U to continue or T to terminate

This message is issued during execution of the VSAM data set loading procedure to prompt the operator to confirm the request.

The ISX operator action: Enter U to confirm or T to reject the request for loading.

ISX065A Do you have to continue? Reply Y or N

This message is issued during the VSAM data set loading procedure to prompt the operator for the next VSAM data set or to terminate the procedure.

The ISX operator action: Enter Y to continue the processing with the next VSAM data set or N to terminate VSAM data sets loading.

ISX066A Enter device type. Reply 3380 or 3390

This message is issued during VSAM data set loading procedure for CKD DASD to prompt operator to enter the CKD device type.

The ISX operator action: Enter 3380 or 3390 emulated device type.

ISX067W OPEN failed

This message is sent to the operator when ISX detects that an ISX load module library cannot be opened.

ISX068W SYSEVENT failed

This message is sent to the operator when ISX detects that the non-swappable mode of ISX execution in zOS cannot be set up.

ISX069A There is no 'devtype' I/O device

This message is sent to the operator when ISX detects that there is no I/O device specified in the command line.

The ISX operator action: Reissue the ISX command with another I/O device, if required.

ISX070W IPL channel program error

This message is issued when any errors are encountered in a channel program during the guest initial program loading process.

ISX071A Invalid IPL device type

This message is issued when the operator is trying to perform an IPL command having specified I/O device that is not supported by ISX.

The ISX operator action: Check the IPL device type.

ISX072A Invalid operand combination

This message is issued if ISX detects an invalid sequence of operands specified on a command line.

The ISX operator action: Check the command syntax.

ISX073W IPL device error. CC=cc CSW=ddss SENSE=xxx...x[...]

This message is issued when any I/O errors are encountered during the guest initial program loading process. CC=cc defines the condition code, CSW=ddss defines the I/O device and subchannel status of the last I/O operation, SENSE=xxx...x defines the sense data.

ISX074A Invalid storage length

This message is issued if the specified data has an invalid length or if the specified guest storage address plus specified data length is out of the guest address space.

The ISX operator action: Check the storage length value.

ISX075A Device xxxx is not allocated to nnnnnnnn for yyyy

This message is issued if the real I/O device xxxx is not allocated by zOS to ISX job nnnnnnnn for the guest I/O device yyyy.

The ISX operator action: Check the I/O device number xxxx, check zOS I/O configuration, check the I/O device allocation in zOS.

ISX076A Device xxxx is out of zOS

This message is issued when ISX cannot receive data about I/O device xxxx from zOS.

The ISX operator action: Check the specified device number.

ISX077A Invalid command - xxxxxxxx

This message is issued when ISX cannot recognize the entered command text xxxxxxxx.

The ISX operator action: Check the ISX command code.

ISX078A Console is already defined

This message is issued when the operator is trying to define the console device using the DEFINE command, but the console device is already defined for the guest I/O configuration.

The ISX operator action: Check the previous console definition.

ISX079A Device xxxx is already defined

This message is issued when the operator is trying to define I/O device xxxx using the DEFINE command, but the device xxxx is already defined for the guest I/O configuration.

The ISX operator action: Check the previous device definition.

ISX080A Invalid guest storage address

This message is issued if the guest storage address specified in a command line is out of the guest interpretive space.

The ISX operator action: Check the specified guest storage address.

ISX081A Interpretive space not available

This message is issued when any operation with the guest storage is required, but the interpretive space storage has not been defined by the SET command.

The ISX operator action: Define the guest storage with the SET command.

ISX082A Device xxxx is already attached

This message is issued when the operator is trying to attach a real I/O device xxxx that is already attached to the guest I/O configuration.

The ISX operator action: Check the previous device attachment.

ISX083W GETMAIN failed

This message is issued during execution of a command processor if the ISX region or SQA of zOS does not have enough free storage for allocation.

ISX084A Operand missing or invalid

This message is issued if ISX cannot recognize the specified operand or a required operand is not specified in the command line.

The ISX operator action: Check the command syntax.

ISX085W LOAD failed for mmmmmm

This message is sent to the operator when ISX detects that an ISX module mmmmmm cannot be loaded by the LOAD macro.

ISX086W IDENTIFY failed

This message is sent to the operator when ISX detects an error on the IDENTIFY macro.

ISX087A There is no subchannel number for device xxxx

This message is issued when the ISX operator is trying to attach or define the guest I/O device xxxx, but there is no free guest subchannel number for that attachment.

The ISX operator action: Check the number of guest I/O devices.

ISX088A Input not accepted

This message is sent to the system operator when the ISX input buffer is busy with a previous input command, or the console device was not defined for the guest system, or specific console was defined with the SET CONSNAME command, but input is from another zOS console.

The ISX operator action: Resubmit the request at a later time or define the guest console device.

ISX089I Guest reading is cancelled

This message is sent to the system console to indicate that the read operation initiated by the guest system for the guest console device is cancelled.

ISX090A Guest is reading

This message is sent to the system console to indicate that the read operation is initiated by the guest system for the guest console device.

The ISX operator action: Enter input data for the guest request.

ISX091A Device xxxx does not exist

This message is issued if an operation is required for the I/O device xxxx, but the device has not been defined in the guest I/O configuration.

The ISX operator action: Check the specified device number.

ISX092I Disable wait PSW = xxxxxxxx xxxxxxxx

This message is issued when ISX detects that the current guest PSW has the I/O and external interrupts system mask turned off and the wait bit turned on. The guest system is placed in a stopped state.

ISX093I Complex E/P interrupt loop. PSW = xxxxxxxx xxxxxxxx

This message is issued, when a complex external/program interrupt loop has been detected during execution of the guest system.

ISX094I External interrupt loop. PSW = xxxxxxxx xxxxxxxx

This message is issued when an external interrupt loop has been detected during execution of the guest system.

ISX095I Program interrupt loop. PSW = xxxxxxxx xxxxxxxx

This message is issued when a program interrupt loop has been detected during execution of the guest system.

ISX096W ISX/cmp initialization failed

This message is issued if errors have been detected during the ISX, ISX/LSA, or ISX/LCS initialization process.

ISX097I Path from xxxx, jjjjjjjj to aaaaaaaaaa 'text'

This message is issued when the data transfer path from the token-ring adapter xxxx of ISX job jjjjjjjj to the token-ring adapter with hardware address of aaaaaaaaaa is allocated, rejected, or failed.

ISX098W ESTAE failed

This message is issued if ISX cannot define its own abnormal end exit routine.

ISX099I ISX/pgm Ver v/Rel r initialization completed

This message is issued when ISX, ISX/LSA, or ISX/LCS Version 'v', Release 'r' initialization has successfully completed.

ISX100A Data set nnn...n is in use

This message is issued when data set nnn...n is requested by ISX for shared or exclusive control, but the data set is not immediately available.

The ISX operator action: Resubmit the request at a later time or use another data set.

ISX101I ISX/cmp is running

This message is issued when token-ring adapter emulation is executed by ISX/LSA or ISX/LCS.

ISX102I Token-ring adapter xxxx, jjjjjjjj is delayed

This message is issued when the emulated token-ring adapter xxxx of ISX job jjjjjjjj is delayed. Increase the priority of the ISX job in zOS.

ISX104A Invalid ppppp for the device

This message is issued when the data set specified on a DEFINE or TAKE command has an invalid RECFM or LRECL for the I/O device emulation.

The ISX operator action: Check the corresponding data set parameter.

ISX105I ISX/cmp is stopped

This message is issued when ISX/LSA or ISX/LCS stops execution of the emulated token-ring adapter.

ISX106I Device xxxx, jjjjjjjj disassociated with [SYSOUT] data set [nnn...n]

This message is issued by ISX job jjjjjjjj, when an unusual I/O condition or end of data condition is detected during emulation of device xxxx associated with a SYSOUT or permanent data set, or the TAKE command was issued with the DSNAME operand, but without data set name for xxxx device. The connection with the data set is revoked.

ISX107A Data set nn...n has insufficient space

This message is issued, when VSAM data set specified in the DEFINE or TAKE command has insufficient size for magnetic tape emulation. The minimal size is 32 blocks.

The ISX operator action: Check the corresponding data set parameter.

ISX109I xxxx, jjjjjjjj display: text1 : text2

This message is issued when the guest system displays data on the display of emulated magnetic tape.

ISX110A cmp is already defined

This message is issued when the DEFINE command is issued by the ISX operator, but parameters of the emulated token-ring adapter are already defined for ISX/LSA or ISX/LCS.

The ISX operator action: Check the previous LSA or LCS definition.

ISX111A cmp is not previously defined

This message is issued when the DEFINE or the SET command is issued by the ISX operator, but parameters of the emulated token-ring adapter have not been previously defined for ISX/LSA or ISX/LCS.

The ISX operator action: Check the previous LSA or LCS definition.

ISX112A ISX/cmp, jjjjjjjj reinstated

This message is issued when the ISX/cmp component of jjjjjjjj job is reinstated. The cmp value defines one of the following components of ISX:

ISX/URD - The Unit Record Device Emulator;

ISX/FBA - The FBA DASD Emulator;

ISX/CKD - The CKD DASD Emulator;

ISX/MTP - The Magnetic Tape Emulator;

ISX/GRF - The Display Unit Emulator;

ISX/CCA - The Channel-To-Channel Adapter Emulator;

ISX/CET - The CET Token-Ring Adapter Emulator;

ISX/LCS - The LCS Token-Ring Adapter Emulator

ISX113W ISX/cmp, jjjjjjjj reinstatement failed

This message is issued when reinstatement of the ISX/nnn component of ISX job jjjjjjjj failed. The cmp value defines one of the following components of ISX:

ISX/URD - The Unit Record Device Emulator;

ISX/FBA - The FBA DASD Emulator;

ISX/CKD - The CKD DASD Emulator;

ISX/MTP - The Magnetic Tape Emulator;

ISX/GRF - The Display Unit Emulator;

ISX/CCA - The Channel-To-Channel Adapter Emulator;

ISX/CET - The CET Token-Ring Adapter Emulator;

ISX/LCS - The LCS Token-Ring Adapter Emulator

ISX115I Prefix from xxxx, jjjjjjjj >> ppppppppppppppppp

This message is issued when a prefix ppppppppppppppppp is sent by the user of the emulated token-ring adapter to ISX/LSA job jjjjjjjj via real CTCA xxxx, or when the prefix is sent by the emulated token-ring adapter xxxx of ISX job jjjjjjjj to real LSA adapter.

ISX116I Prefix to xxxx, jjjjjjjj << ppppppppppppppppp

This message is issued when a prefix ppppppppppppppppp is received by the user of the emulated token-ring adapter with ISX/LSA job name jjjjjjjj via real CTCA xxxx, or when the prefix is received by the emulated token-ring adapter xxxx of ISX job jjjjjjjj from real LSA adapter.

ISX118W Unknown frame to xxxxxxxxxxxx, jjjjjjjj

This message is issued when a frame of unknown type is received by ISX/LSA or ISX/LCS for the emulated token-ring adapter xxxxxxxxxxxx of ISX/LSA or ISX/LCS job jjjjjjjj.

ISX119I Frame to xxxxxxxxxxxx, jjjjjjjj is discarded

This message is issued when a frame is received by ISX/LSA or ISX/LCS for the emulated token-ring adapter xxxxxxxxxxxx of ISX/LSA or ISX/LCS job jjjjjjjj, but was discarded.

ISX120W Invalid frame to xxxxxxxxxxxx, jjjjjjjj << ffffffff

This message is issued when an invalid frame is received by the emulated token-ring adapter xxxxxxxxxxxx with ISX/LSA or ISX/LCS job name jjjjjjjj from the virtual token-ring.

ISX121W Invalid primitive to xxxx, jjjjjjjj << ffffffff ffffffff

This message is issued when invalid primitive ffffffff ffffffff is received by the emulated token-ring adapter with ISX/LSA job name jjjjjjjj from the user of the token-ring adapter via real CTCA xxxx.

ISX123I Primitive to xxxx, jjjjjjjj is discarded

This message is issued when a primitive is received by CTCA xxxx of the emulated token-ring adapter with ISX/LSA job name jjjjjjjj, but was discarded.

ISX124W Unknown primitive from xxxx, jjjjjjjj << ppppppppp ppppppppp

This message is issued when a primitive of unknown type ppppppppp ppppppppp is sent by the user of the emulated token-ring adapter to ISX/LSA job jjjjjjjj via real CTCA xxxx.

ISX125I Primitive from xxxx, jjjjjjjj >> ppppppppppppppppp...

This message is issued when an outbound primitive ppppppppppppppppp... is sent by the user of the emulated token-ring adapter to ISX/LSA job jjjjjjjj via real CTCA xxxx, or when the outbound primitive is sent by the emulated token-ring adapter xxxx of ISX job jjjjjjjj to real LSA adapter.

ISX126I Primitive to xxxx, jjjjjjj << pppppppppppppppp...

This message is issued when an inbound primitive pppppppppppppppp... is received by the user of the emulated token-ring adapter with ISX/LSA job name jjjjjjj via real CTCA xxxx, or when the inbound primitive is received by the emulated token-ring adapter xxxx of ISX job jjjjjjj from real LSA adapter.

ISX127I Frame to xxxxxxxxxxxx, jjjjjjj << fffffff...

This message is issued when an inbound frame fffffff... is received by the emulated token-ring adapter xxxxxxxxxxxx with ISX/LSA or ISX/LCS job name jjjjjjj from the virtual token-ring.

ISX128I Frame from xxxxxxxxxxxx, jjjjjjj >> fffffff...

This message is issued when an outbound frame fffffff... is sent by the emulated token-ring adapter xxxxxxxxxxxx with ISX/LSA or ISX/LCS job name jjjjjjj to the virtual token-ring.

ISX129I Path to ssssssssss of jjjjjjj from aaaaaaaaaa text

This message is issued when a data transfer path to the token-ring station ssssssssss of ISX/LSA or ISX/LCS job jjjjjjj from the token-ring adapter with hardware address of aaaaaaaaaa is rejected or failed.

ISX130I Path from ssssssssss of jjjjjjj to aaaaaaaaaa text

This message is issued when a data transfer path from the token-ring station ssssssssss of ISX/LSA or ISX/LCS job jjjjjjj to the token-ring adapter aaaaaaaaaa is set up, rejected, or failed.

ISX131I ccc connections is allocated for sap SAP of xxxx, jjjjjjj

or

ISX131I ccc connections is not allocated for sap SAP of xxxx, jjjjjjj

This message is issued when allocation of ccc connections for sap service access point was requested for token-ring adapter xxxx of ISX job jjjjjjj. Depending on the maximum connections value of the corresponding real LSA adapter, this message defines whether the allocation request is accepted or not.

ISX132A Maximum connections exceeded for sap SAP of xxxx, jjjjjjj

This message is issued when the maximum number of connections for the sap service access point was exceeded for token-ring adapter xxxx of ISX job jjjjjjj.

The ISX operator action: Check the maximum connections of the corresponding real LSA adapter is not exceeded. If the value is not exceeded, then deactivate sap SAP of the guest software and redefine the allocation of ccc connections for sap SAP of the guest software for the emulated token-ring adapter with the guest device number of xxxx using the TAKE command, and, farther, activate sap SAP of the guest software.

ISX133A sap SAP of xxxx, jjjjjjj is active

This message is issued when the ISX operator is trying to define a new number of connections for sap service access point of token-ring adapter xxxx of ISX job jjjjjjj, but the specified service access point is already activated.

The ISX operator action: Deactivate sap SAP of the guest software and redefine the allocation of ccc connections for sap SAP of the guest software for the emulated token-ring adapter with the guest device number of devaddr using the TAKE command, and, farther, activate sap SAP of the guest software.

ISX134A Too much connections requested for sap SAP of xxxx, jjjjjjj

This message is issued when too many connections are requested for the sap service access point for token-ring adapter xxxx of ISX job jjjjjjj.

The ISX operator action: Depending on the maximum connections of the corresponding real LSA adapter redefine the allocation of ccc connections for sap SAP of the guest software for the emulated token-ring adapter with the guest device number of devaddr using the TAKE command.

ISX136A ISX/pgm is stopped at address xxxxxxxx

This message is issued, when ISX, ISX/LSA, or ISX/LCS is stopped by request of the XSTOP command.

The ISX operator action: Perform an action on your choice and, further, issue the XBEGIN command to continue or the SHUTDOWN command to terminate ISX processing.

ISX137I ISX/pgm is running

This message is issued, when ISX, ISX/LSA, or ISX/LCS is running by request of the XBEGIN command.

ISX138W MAXIMUM OF GUEST SYSTEMS EXHAUSTED

This message is issued during execution of the IPL command, when maximum of guest systems exhausted.

The ISX operator action: Do not IPL more than the number of guest systems defined via the MAXGUEST operand of the SET command.