# BEA CRM Administration Guide

**BEA CRM Administration Guide**

| Document Edition | Part Number | Date | Software Version |
| --- | --- | --- | --- |
| 4.0 | N/A | January 2001 | BEA Communications Resource Manager 4.0 |

# Contents

## 3. Administering the SNA Components

## A. Command Reference Pages

## B. Error Messages

## Glossary

## Index

# About This Document

This document explains how the BEA Communications Resource Manager (CRM) provides the emulation allowing Customer Information Control System (CICS) and Information Management System (IMS) protocols to flow into and out of theTuxedo environment. This document also describes how to administer application domains with the CRM.

This document covers the following topics:

- "Understanding the Communications Resource Manager," describes the CRM and provides information about remote host domain configurations and sample Virtual Telecommunications Access Method (VTAM) configurations.

- "Sample Configurations and JCL," describes some sample VTAM configurations and sample JCL for running the CRM commands on an MVS operating system.

- "Administering the SNA Components," describes how to use the CRM for administering communications resources.

- "Command Reference Pages," lists and describes CRM system commands.

- "Error Messages," describes CRM error messages.

- Glossary

# What You Need to Know

This document is intended mainly for system administrators and operators who will use the CRM to monitor and link communications resources between mainframe and UNIX or Windows NT applications.

# How to Use The Documentation

The Documentation CD included in the package with your product software CD contains an HTML Web User Interface (WUI). The WUI links to HTML versions and PDF versions of eLink Adapter for Mainframe (eAM) documentation, along with this guide. The WUI should be viewed in an online browser. The PDF versions should be used for printing. (Information on how to view the online documentation is available in the release notes accompanying your product software.)

**Note:** The WUI requires a Web browser that supports HTML 3.0, along with Netscape Navigator 4.0 or later, or Microsoft Internet Explorer 4.0 or later.

You must have the Adobe Acrobat Reader to print the PDF file. If you do not have this reader, you can obtain it free of charge from the Adobe Systems Incorporated home site at `www.adobe.com`. (The WUI contains a hot link to this site.)

# e-docs Web Site

BEA product documentation is available on the BEA corporate Web site. From the BEA Home page, click on Product Documentation or go directly to the "e-docs" Product Documentation page at http://e-docs.beasys.com.

# Related Information

The following documents contain information that is relevant to using the CRM:

- *BEA eLink Adapter for Mainframe Installation Guide Version 4.0*

- *BEA eLink Adapter for Mainframe User Guide Version 4.0*

- *BEA eLink Adapter for Mainframe Release Notes Version 4.0*

# Documentation Conventions

The following documentation conventions are used throughout this document:

| Convention | Item |
|---|---|
| **boldface text** | Indicates terms defined in the glossary. |
| blue text | Indicates hypertext links in PDF documents. |
| Ctrl+Tab | Indicates that you must press two or more keys simultaneously. |
| *italics* | Indicates emphasis or book titles. |
| `monospace text` | Indicates code samples, commands and their options, data structures and their members, data types, directories, and file names and their extensions. Monospace text also indicates text that you must enter from the keyboard. *Examples*: <br> `#include <iostream.h> void main ( ) the pointer psz` <br> `chmod u+w *` <br> `\tux\data\ap` <br> `.doc` <br> `tux.doc` <br> `BITMAP` <br> `float` |
| `monospace boldface text` | Identifies significant words in code. *Example*: <br> `void` **`commit`** `( )` |
| `monospace italic text` | Identifies variables in code. *Example*: <br> `String` *`expr`* |

| Convention | Item |
|---|---|
| UPPERCASE TEXT | Indicates device names, environment variables, and logical operators.<br>*Example*s:<br>LPT1<br>SIGNON<br>OR |
| { } | Indicates a set of choices in a syntax line. The braces themselves should never be typed. |
| [ ] | Indicates optional items in a syntax line. The brackets themselves should never be typed.<br>*Example*:<br>`buildobjclient [-v] [-o name ] [-f file-list]...`<br>`[-l file-list]...` |
| \| | Separates mutually exclusive choices in a syntax line. The symbol itself should never be typed. |
| ... | Indicates one of the following in a command line:<br>■ That an argument can be repeated several times in a command line<br>■ That the statement omits additional optional arguments<br>■ That you can enter additional parameters, values, or other information<br>The ellipsis itself should never be typed.<br>*Example*:<br>`buildobjclient [-v] [-o name ] [-f file-list]...`<br>`[-l file-list]...` |
| .<br>.<br>. | Indicates the omission of items from a code example or from a syntax line. The vertical ellipsis itself should never be typed. |

# Contact Us

Your feedback on the BEA CRM Administration Guide is important to us. Send us e-mail at **docsupport@bea.com** if you have questions or comments. Your comments will be reviewed directly by the BEA professionals who create and update the CRM documentation.

In your e-mail message, please indicate that you are using the documentation for the Communications Resource Manager, Document Edition 1.0.

If you have any questions about this version of the eLink Adapter for Mainframe, or if you have problems installing and running the CRM, contact BEA Customer Support through BEA WebSupport at **www.bea.com**. You can also contact Customer Support by using the contact information provided on the Customer Support Card, which is included in the product package.

When contacting Customer Support, be prepared to provide the following information:

- Your name, e-mail address, phone number, and fax number

- Your company name and company address

- Your machine type and authorization codes

- The name and version of the product you are using

- A description of the problem and the content of pertinent error messages

# 1 Understanding the Communications Resource Manager

This section discusses the following topics:

- About the Communications Resource Manager

- System Configuration

- Remote Host Domain Configuration

**Note:** All references to ATMI files, functions, and documentation apply to Tuxedo, eLink Platform, and WebLogic Enterprise files, functions, and documentation.

## About the Communications Resource Manager

The Communications Resource Manager (CRM) is the component of the BEA eLink Adapter for Mainframe (eAM) that manages communications resources. The CRM coordinates the flow of data between applications running on an ATMI platform and applications running on a mainframe. The mainframe applications may use the following protocols:

- Customer Information Control System/Enterprise System Architecture (CICS/ESA)
  - Distributed Transaction Processing (DTP)
  - Distributed Program Link (DPL)
- Information Management System (IMS)

The CRM runs as a separate native process providing emulation that allows CICS/ESA and IMS protocols to flow into and out of the ATMI environment.

The CRM must run on the same platform as the SNA stack, but it may run on a different platform from the ATMI system and the eAM gateway (GWSNAX). The eAM gateway provides the configuration for the CRM. If the eAM gateway is to be brought up on a platform other than the one the CRM is on, then the CRM should already be started and monitoring the address specified in the eAM gateway configuration.

# System Configuration

The CRM system may be configured as either a local configuration or a distributed configuration running on UNIX, Windows NT, or a mainframe operating system. For a complete list of operating systems, refer to the *BEA eLink Adapter for Mainframe Release Notes*. If the CRM is not run on a mainframe, it must run on the same platform as the SNA stack. If the eAM gateway is to be brought up on a platform other than the one the CRM is on, then the CRM should already be started and monitoring the address specified in the eAM gateway configuration.

# Local Configuration

The local configuration, illustrated in Figure 1-1, combines the applications, eAM gateway, ATMI platform, and the CRM with the stack (PU2.1 server) on the same UNIX or Windows NT platform. It employs the IBM proprietary SNA protocol for communication with the mainframe via the stack.

**Figure 1-1   eAM Local Configuration**
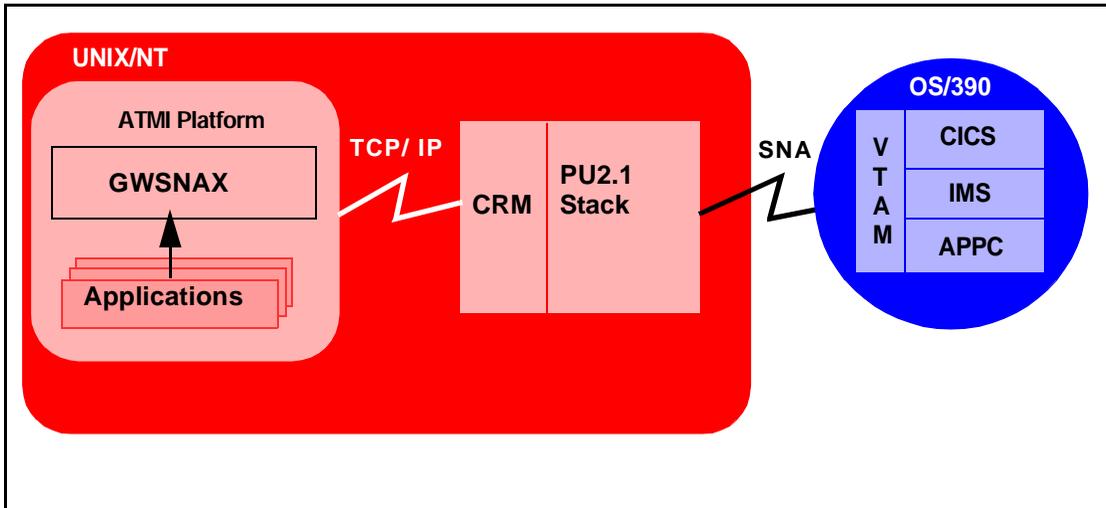
# Distributed Configurations

One type of distributed configuration separates the applications and the eAM gateway on a UNIX or Windows NT platform from the CRM by installing the CRM on the IBM OS/390 Mainframe. See Figure 1-2. This configuration eliminates the need for a third-party stack on the UNIX or NT machine. Note that this configuration requires a one-to-one relationship between the local eAM gateway and the remote CRM.

**Figure 1-2   eAM Distributed Configuration**
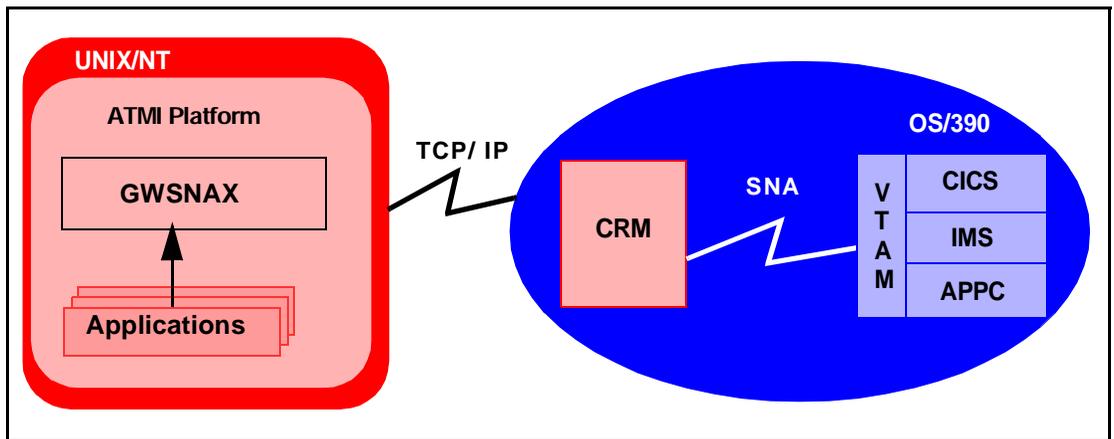


Another type of distributed configuration separates the applications and the eAM gateway from the CRM on different UNIX or Windows NT platforms. See Figure 1-3. This configuration employs Transmission Control Protocol/Internet Protocol (TCP/IP) connectivity between the applications platform and the CRM platforms, as well as the SNA connectivity to the mainframe environment(s). It provides the flexibility to deploy the ATMI platform separately from the CRM with installations that require the

ATMI platform on a platform other than the one on which the SNA stack is running. Note that this configuration requires a one-to-one relationship between the local eAM gateway and the remote CRM.

**Figure 1-3   Alternate eAM Distributed Configuration**
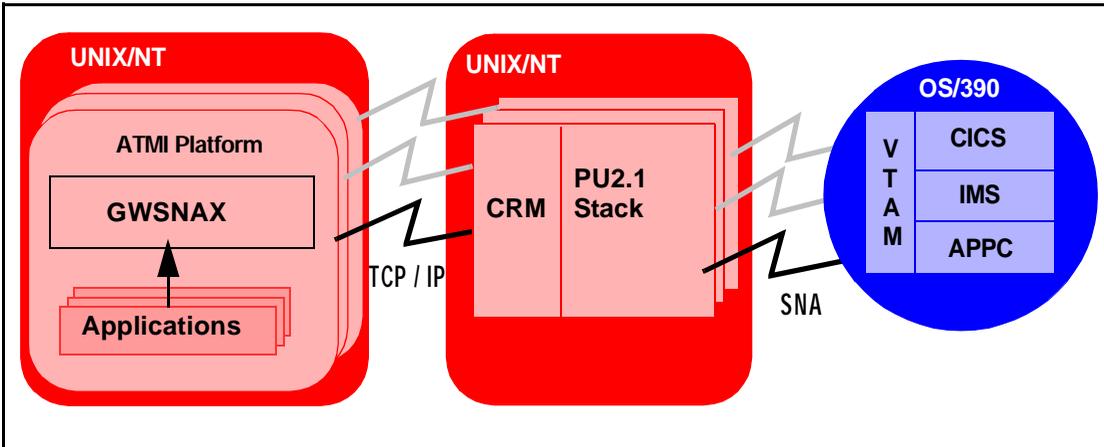


# Remote Host Domain Configuration

A basic understanding of the mainframe configuration requirements provides a context for understanding the CRM functions and configuration requirements.

**Note:**   Consult with your local mainframe system administrator for specific information about your system. Any samples provided illustrate a starting point for configuring your system and do not represent all possibilities. The samples represent one way a mainframe can be configured to work in an Advanced Peer-to-Peer Networking (APPN) Local Area Network (LAN) environment.

Ensure that the CICS/ESA remote domain is prepared to conduct operations with the BEA local domain by:

■   Establishing the VTAM Configuration

■   Configuring the CICS/ESA LU

- Creating Connections at the Remote Host
- Defining the Session at the Remote Host

■ Completing Cross-Platform Definitions

# Establishing the VTAM Configuration

If your eAM system is used in a Virtual Telecommunications Access Method (VTAM) environment, make sure the host configuration supports it. Refer to "Sample Configurations" for examples based on the requirements for the eAM.

# Configuring the CICS/ESA LU

Before you can connect the CRM to the remote stack, the CICS/ESA LU (Logical Unit) configuration must be established. To accomplish this task, you must create connection definitions, create session definitions, and install resource definitions.

## Creating Connections at the Remote Host

If a remote connection definition file is not already in place, you must work with the mainframe support personnel to create one. When placed on the remote host, the definition provides a connection with the local domain. Note the following example of an eAM connection definition file:

```
DEFINE CONNECTION(BEA)        GROUP(BEACONN)
       DE(eAM EXAMPLE RDO CONNECTION)
       ACCESSMETHOD(VTAM)     PROTOCOL(APPC)
       NETNAME(**VTAM NETWORK NAME OF REMOTE SYSTEM**)
       ATTACHSEC(LOCAL)       AUTOCONNECT(NO)
```

To install the sample connection definition, put it on the host in a separate group. Use the CEDA INSTALL command.

For example:

```
CEDA INSTALL GROUP(BEACONN)
```

## Defining the Session at the Remote Host

If a session definition is not already in place, work with the mainframe support personnel to create one. When placed on the remote host, the session definition defines the logical links by which the local domain communicates with the remote host. Note the following example of an eAM session definition:

```
DEFINE SESSION(BEATEST)        GROUP(BEACONN)
       CONNECTION(BEA)
       DE(eAM EXAMPLE RDO SESSION)
       PROTOCOL(APPC)          AUTOCONNECT(YES)
       MODENAME(**MODE**)      MAXIMUM(**SESSNBR**,**WINNER**)
```

The arguments and options in this example are defined in the following way:

AUTOCONNECT

Indicates how the activation of the session is negotiated.

YES

Enables the CICS/ESA host to negotiate its own winner sessions when a conversation is allocated.

MODENAME

Indicates either a CICS/ESA-supplied mode name, such as SMSNA100, or your own defined mode name. If another set of session definitions exists for the BEA connection, this mode name must be unique among all sets defined to the connection. The mode name corresponds to the VTAM LOGMODE name.

MAXIMUM

Defines the total number of sessions in the set and the total number of winner sessions. The total number of winner sessions must include those for the host and the remote stack. The WINNER number plus the remote WINNER number should equal the SESSNBR.

## Viewing Connection and Session Status

After you have installed group definitions, you can view the status of connections and sessions using the following CICS/ESA system commands:

```
CEMT I CONN(BEA)            **view the status of the connection
CEMT I NET(**Netname**)     **View the status of the sessions
CEMT I MODENAME(**MODE**)   **View the status of the mode
```

# Completing Cross-Platform Definitions

Consult with your CICS/ESA remote domain administrator to obtain key parameters in the VTAM definition that must be included in the SNA stack configuration, as well as in other configuration files in the eAM local domain.

## VTAM Cross-Platform Definitions

Before installing eAM software, please review Table 1-1 for a summary of cross-platform definitions. Consult with the VTAM system administrator to obtain the value indicated in the *Name* column and make the corresponding entries shown in the *Needed In* column.

DCL-based stacks referred to in Table 1-1 include the following stacks:

- HP SNAplus2

- Solaris SNAP-1X

- IBM Comm Server

**Table 1-1  Summary of Cross-Platform Definitions**

| Name | Originates In | Needed In |
|------|---------------|-----------|
| SNA Network ID (e.g. **SNANET1**)<br><br>and VTAM Host ID (e.g. **VTAMHOST**) | VTAM configuration | **SunLink SNA Stack Configuration:**<br>Example:<br>`CP    NQ_CP_NAME=`**`SNANET1`**`.SPARC1`<br>`DLC    RMTNQ_CP_NAME=`**`SNANET1.VTAMHOST`**<br>`LU    NQ_LU_NAME=`**`SNANET1`**`.L0P0024A`<br>`PTNR_LU    NQ_LU_NAME=`**`SNANET1`**`.CICSSYN`<br>**DCL-based Stack Configuration:**<br>Example:<br>`fqcp_name=`**`SNANET1`**`.SPARC1`<br>`adj_cp_name=`**`SNANET1.VTAMHOST`**<br>`fqplu_name=`**`SNANET1`**`.CICSSYN` |

**Table 1-1  Summary of Cross-Platform Definitions**

| Name | Originates In | Needed In |
|------|--------------|-----------|
| Mode Name (e.g. **SNA62**) | VTAM-MODEENT definition | **CICS Sessions Definition:**<br>Example:<br>`MODENAME(`**`SNA62`**`)`<br><br>**SunLink SNA Stack Configuration:**<br>Example:<br>`MODE    NAME=`**`SNA62`**<br><br>**DCL-based Stack Configuration:**<br>Example:<br>`mode_name=`**`SNA62`**<br><br>**GWSNAX Configuration:**<br>Example:<br>`*DM_SNA_LINKS`<br>`MODENAME="`**`SNA62`**`"` |
| Control Point Name CPNAME (e.g. **SPARC1**) | VTAM-PU definition | **SunLink SNA Stack Configuration:**<br>Example:<br>`CP    NAME=`**`SPARC1`**<br>`CP    NQ_CP_NAME=SNANET1.`**`SPARC1`**<br><br>**DCL-based Stack Configuration:**<br>Example:<br>`fqcp_name=SNANET1.SPARC`<br>`cd_alias=`**`SPARC1`** |
| Local LU Name (e.g. **L0F0024A**) | VTAM-LU definition | **CICS CONNECTION definition:**<br>Example:<br>`NETNAME(`**`L0F0024A`**`)`<br>**SunLink SNA Stack Configuration:**<br>Example:<br>`LU    NAME=`**`L0F0024A`**<br>`LU    NQ_LU_NAME=SNANET1.`**`L0F0024A`**<br>`PTNR_LU    LOC_LU_NAME=`**`L0F0024A`**<br>`TP    LOC_LU_NAME=`**`L0F0024A`**<br><br>**DCL-based Stack Configuration:**<br>Example:<br>`lu_name=`**`L0F0024A`**<br>`lu_alias=`**`L0F0024A`**<br><br>**GWSNAX Configuration:**<br>Example:<br>`*DM_SNA_STACKS`<br>`LOCALLU="`**`L0F0024A`**`"` |

**Table 1-1  Summary of Cross-Platform Definitions**

| Name | Originates In | Needed In |
|------|---------------|-----------|
| CICS LU Name (e.g. **CICSSYN**) | VTAM-LU definition | **SunLink SNA Stack Configuration:**<br>Example:<br>`PTNR_LU    NAME=`**`CICSSYN`**<br>`PTNR_LU    NQ_LU_NAME=SNANET1.`**`CICSSYN`**<br>`MODE    PTNR_LU_NAME=`**`CICSSYN`**<br><br>**DCL-based Stack Configuration:**<br>Example:<br>`fqplu_name=SNANET1.`**`CICSSYN`**<br>`plu_alias=`**`CICSSYN`**<br><br>**GWSNAX Configuration:**<br>Example:<br>`*DM_SNA_LINKS`<br>`RLUNAME="`**`CICSSYN`**`"` |
| Terminal Identifier (e.g. **05DF0024**) | VTAM (IDNUM+IDBLK) | **SunLink SNA Stack Configuration:**<br>Example:<br>`DLC    TERM << ALIGN=CHAR`<br><br>**DCL-based Stack Configuration:**<br>Example:<br>`node_id=<`**`05000002`**`>` |
| Local Network Device (e.g. **/dev/tr**) | UNIX Configuration | **SunLink SNA Stack Configuration:**<br>Example:<br>`TRLINE    DEVICE='`**`/dev/tr`**`'`<br>or<br>`SDLCLINE    DEVICE='`**`/dev/dcp1`**`'` |
| Local MAC Address (token ring only) | Token ring address of Host | **SunLink SNA Stack Configuration:**<br>Example:<br>`TRLINE    Source_Address=X'080020117d7a'` |
| Remote MAC Address (token ring only) | Token ring address of local machine | **SunLink SNA Stack Configuration:**<br>Example:<br>`DLC    RMTMACADDR=X'40000101000'`<br><br>**DCL-based Stack Configuration:**<br>Example:<br>`mac_address=<400031720001>` |
| LAN Speed (e.g. **4MBs**) | Speed of token ring network | **SunLink SNA Stack Configuration:**<br>Example:<br>`TRLINE    LAN_rate=RING_`**`4MBS`** |

**Table 1-1  Summary of Cross-Platform Definitions**

| Name | Originates In | Needed In |
|------|---------------|-----------|
| SDLC parameters (line protocol) | VTAM-line definition | **SunLink SNA Stack Configuration:**<br>Example:<br>`SDLCLINE    DUPLEX=half`<br>`LINE=switched`<br>`NRZI=no`<br>`PAUSE=1`<br>`SPEED=4800` |
| Partner Definition (e.g. **TUXPART1**) | CICS/ESA | **Mainframe Client Application:**<br>(for example, COBOL with embedded CPI-C to route CICS to appropriate LU for BEA Connect SNA)<br>COBOL Example:<br>`MOVE 'TUXPART1' TO SYM-DEST-NAME`<br>`CALL 'CMINIT' USING CONVERSATION-ID`<br>`        SYM-DEST-NAME, CM_RETCODE`<br>`END-CALL` |
| Set LU definition so maximum sync-level allowed corresponds to DMCONFIG file entry: *DM_SNA_LINKS MAXSYNCLVL | Stacks | **SunLink SNA Stack Configuration:**<br>Example:<br>`SYNC_LVL=SYNCPT`<br>**DCL-based Stack Configuration:**<br>Example:<br>`[define local_lu]`<br>`Syncpt._Support=NO` |
| Map all incoming conversations to eAM gateway (make sure TPs have all privileges available, e.g. CNOS, syncpoint if licensed, service conversations, etc.). | Stacks | **SunLink SNA Stack Configuration:**<br>Example:<br>`TP_HEXNAME=x'2a'`<br>**DCL-based Stack Configuration:**<br>Example:<br>`Sna_tps`<br>`<404040...hex representation of 64`<br>`EBCDIC spaces...404040>`<br>`TYPE=QUEUED`<br>`TIMEOUT=-1`<br>`USERID=authorized_user_here`<br>`GROUP=authroized_group_here`<br>`LUALIAS=lu_name_here` |

**Table 1-1  Summary of Cross-Platform Definitions**

| Name | Originates In | Needed In |
|------|---------------|-----------|
| CICS Transaction IDs (e.g. TOUP) | CICS/ESA | **GWSNAX Configuration:**<br>Example:<br>`*DM_REMOTE_SERVICES` |

## OS/390 UNIX Platform Definitions

Before installing eAM software, please review Table 1-2 for a summary of SNA definitions when CRM runs on an OS/390 UNIX platform. Consult with the system administrator to obtain the value indicated in the *Name* column and make the corresponding entries shown in the *Needed In* column.

**Table 1-2  Summary of OS/390 SNA Definitions**

| Name | Originates In | Needed In |
|------|---------------|-----------|
| Local LU Name (e.g. **BEAAPPL1**) | VTAM-LU definition | **CICS CONNECTION definition:**<br>Example:<br>`NETNAME(`**`BEAAPPL1`**`)`<br><br>**VTAM Configuration:**<br>Example:<br>`BEASNA VBUILD TYPE=APPL`<br>`BEAAPPL1 APPL ACB=BEAAPPL1,`<br>`            APPC=YES,`<br>`            PARSESS=YES`<br><br>**GWSNAX Configuration:**<br>Example:<br>`*DM_SNA_STACKS`<br>`LOCALLU="`**`BEAAPPL1`**`"` |

**Table 1-2  Summary of OS/390 SNA Definitions**

| Name | Originates In | Needed In |
|---|---|---|
| Mode Name (e.g. **SNA62**) | VTAM-MODEENT definition | **CICS Sessions Definition:**<br>Example:<br>`MODENAME(`**`SNA62`**`)`<br><br>**GWSNAX Configuration:**<br>Example:<br>`*DM_SNA_LINKS`<br>`MODENAME="`**`SNA62`**`"`<br><br>**VTAM Configuration (not required):**<br>Example:<br>`MODEENT=`**`SNA62`** |
| CICS LU Name (e.g. **CICSSYN**) | VTAM-LU definition | **GWSNAX Configuration:**<br>Example:<br>`*DM_SNA_LINKS`<br>`RLUNAME="`**`CICSSYN`**`"` |
| Maximum sync-level allowed | Stacks | **VTAM Configuration:**<br>Example:<br>`SYNCLVL=SYNCPT`<br>**GWSNAX Configuration:**<br>Example:<br>`*DM_SNA_LINKS`<br>`MAXSYSCLVL=2` |
| CICS Transaction IDs (e.g. TOUP) | CICS/ESA | **GWSNAX Configuration:**<br>Example:<br>`*DM_REMOTE_SERVICES` |

## Microsoft SNA Cross-Platform Definitions

Be sure to communicate with the administrator of the CICS/ESA remote domain to obtain key parameters in the VTAM definition that must be included in the Microsoft SNA Server configuration, as well as in other configuration files in the eAM local domain.

Before installing eAM software, please examine the following general procedure for configuring the Microsoft SNA Server. Use the Microsoft SNA Server Manager. Sample values are shown in parenthesis. Consult with the VTAM system administrator to obtain the proper values.

1. Start Microsoft SNA Server Manager from Start on the Task Bar.

2. When a server is automatically created (MVSNT1), note the configuration values displayed in the Server Properties window:
     Server: MVSNT1
     Subdomain: MVSNT1
     Server Role: Primary
     Network Transports: TCP/IP

3. Under Link Services, define a link service (SNADLC1):

   In the Link Service Properties, define DLC 802.2 Link Service Configuration:
     Title: DLC 802.2 Link Service #1
     Adapter: <your ethernet adapter>
     Local Service Access Point (SAP): 0x4
     Use Fixed SAP

4. Under SNA Service Connections, define an 802.2 connection (MVSNT1):

   In the MVSNT1 Properties, define:

   General
     Name: MVSNT1
     Link Service: SnaDlc1
     Remote End: Peer System
     Allowed Directions: Both Directions
     Activation: On Server Startup
     Supports Dynamic Remote APPC LU Definition

     Address
     Remote Network Address: <host MAC address>
     Remote SAP Address: <host SAP address>

   System Identification
     Local Node Name
     Network Name: <mynetwork>
     Control Point Name: MVSNT1
     Local Node ID: <xxx nnnn>XID Type: Format 3
     Remote Node Name

Network Name: <hostnetwork>
Control Point Name: <vtamcpname>
Remote Node ID: Peer DLC Role: Negotiable
Compression Type: None

802.2 DLC
Take Defaults

5. Under Local APPC LUs (SNA Service: Connections: Insert: APPC: Local LU), define a local LU (LUNT1A) in the LUNT1A Properties:

General
LU Alias: LUNT1A
Network Name: <mynetwork>
LU Name: LUNT1A

Advanced
Take Defaults

6. Under Remote APPC LUs, define a remote LU (CICS1) in the CICS1 Properties:

General
Connection: MVSNT1
LU Alias: CICS1
Network Name: <hostnetwork>
LU Name: CICS1
Uninterpreted Name: CICS1

Options
Take Defaults

7. Under APPC Modes, define a mode (SMSNA100) in the SMSNA100 Properties:

General
Mode Name: SMSNA100

Limits
Parallel Session Limit: <max sessions>
Minimum Winner Contention Limit: <min winners>
Partner Min Winner Contention Limit: <max sessions - min winners>
Automatic Activation Limit: 0

Characteristics
Take Defaults

Partners
> Add partnership for Server Name: MVSNT1 between Local LU: LUNT1A and Partner LU: CICS1

Compression
> Take Defaults

# 2 Sample Configurations and JCL

This section discusses the following topics:

# Sample Configurations

The following sections provide sample environments that show how eAM software can be configured for use with an Ethernet LAN and an APPN System 390. Considerations for token ring and subarea-style configurations are included. These samples assume that hardware and operating system installations have been completed.

A properly configured eAM system involves two general types of environments:

- Local environment
  - eAM gateway (GWSNAX)
  - CRM
  - Associated Stacks
- Remote environment
  - Mainframe

# Local Environment

The eAM software is a fully bidirectional program, supporting the local system as either a client or server. This environment consists of the following components:

- Hardware that consists of any workstation and network interface supported by the required software
- Platform operating systems with protocol stacks (PU 2.1 servers)

## Sample Environments

The following samples of a local environment illustrate a starting point when first configuring your system. The attributes of the sample environment machines are listed for reference. Sample configurations are presented for example only and may contain components that are not supported for your system. These samples are not intended to be used without modifications.

**Table 2-1  SPARCstation 5**

| Name | Attribute |
|------|-----------|
| OS | Solaris 2.7 |
| SNA | SunLink 9.1 |
| APP | Tuxedo 6.5 |
| MAC | 08:00:20:7C:47:50 |

**Table 2-1  SPARCstation 5**

| Name | Attribute |
| --- | --- |
| IP | 206.189.43.14 |
| NAME | dalqasun1 |

**Table 2-2  SPARCstation 5**

| Name | Attribute |
| --- | --- |
| OS | Solaris 2.8 |
| SNA | SunLink 9.1 |
| APP | WebLogic Enterprise 5.1 |
| MAC | 08:00:20:87:47:2d |
| IP | 206.189.43.54 |
| NAME | dalqasun1 |

**Table 2-3  HP 9000/847**

| Name | Attribute |
| --- | --- |
| OS | HP-UX B.11.00 (patches:PHNE_9663,9761,9889) |
| SNA | HPSNAPlus2 6.0 |
| APP | Tuxedo 6.5 |
| MAC | 08:00:09:30:24:77 |
| IP | 206.189.43.13 |
| NAME | dalhp10 |

**Table 2-4  Windows NT Server**

| Name | Attribute |
| --- | --- |
| OS | Windows NT Server 4.0, SP2 |
| SNA | MS SNA Server 4.0, SP3 |
| APP | CICS 4.1 |
| MAC | 10:00:5a:d4:c1:e0 |
| IP | 206.189.43.99 |
| NAME | dalnt |

**Table 2-5  P390 Server 500**

| Name | Attribute |
| --- | --- |
| OS | MVS 5.22 9510 |
| SNA | VTAM 4.3 |
| APP | CICS 4.1 / IMS DC 5.1 |
| MAC | 10:00:5a:d4:3e:8e |
| IP | 206.189.43.98 |
| NAME | beav5 |

**Table 2-6  P390 Server 500**

| Name | Attribute |
| --- | --- |
| OS | OS/390 1.2 |
| SNA | VTAM 4.3 |
| APP | CICS 4.1 |
| MAC | 10:00:5a:d4:c1:e0 |
| IP | 206.189.43.96 |

**Table 2-6  P390 Server 500**

| Name | Attribute |
|------|-----------|
| NAME | dalvs7 |

# Local Environment Configurations

## HPSNAPlus2 Configuration

HPSNAPlus2 configurations are usually set up using the HP xSnapAdmin utility, resulting in the configuration text file `/etc/opt/sna/sna_node.cfg`. This file can be manually created and maintained using a text editor; however, using the HP `xsnapadmin` utility is recommended. The following example is the `sna_node.cfg` file for the sample environment.

**Listing 2-1   HPSNAPlus2 Configuration**

```
[define_node_config_file]
major_version = 5
minor_version = 1
update_release = 1
revision_level = 116

[define_node]
node_name = dalhp10
description = snacrm development
node_type = END_NODE
fqcp_name = BEALAN.DALHP10
cp_alias = dalhp10
mode_to_cos_map_supp = NO
mds_supported = YES
node_id = <05ffffff>
max_locates = 100
 dir_cache_size = 255
max_dir_entries = 0
locate_timeout = 60
reg_with_nn = YES
reg_with_cds = YES
mds_send_alert_q_size = 100
cos_cache_size = 24
```

```
tree_cache_size = 40
tree_cache_use_limit = 40
max_tdm_nodes = 0
max_tdm_tgs = 0
max_isr_sessions = 1000
isr_sessions_upper_threshold = 900
isr_sessions_lower_threshold = 800
isr_max_ru_size = 16384
isr_rcv_pac_window = 8
store_endpt_rscvs = NO
store_isr_rscvs = NO
store_dlur_rscvs = NO
dlur_support = YES
pu_conc_support = NO
nn_rar = 128
ptf_flags = NONE

[define_ethernet_dlc]
dlc_name = ETHER0
description = ""
neg_ls_supp = YES
initially_active = NO
adapter_number = 0

[define_ethernet_port]
port_name = ethl0
description = 1st ethernet adapter
dlc_name = ETHER0
port_type = PORT_SATF
port_number = 1
max_rcv_btu_size = 1033
tot_link_act_lim = 64
inb_link_act_lim = 0
out_link_act_lim = 0
ls_role = LS_NEG
act_xid_exchange_limit = 9
nonact_xid_exchange_limit = 5
ls_xmit_rcv_cap = LS_TWS
max_ifrm_rcvd = 7
target_pacing_count = 7
max_send_btu_size = 1033
mac_address = <000000000000>
 lsap_address = 0x08
implicit_cp_cp_sess_support = NO
implicit_limited_resource = NO
implicit_deact_timer = 0
effect_cap = 3993600
connect_cost = 0
byte_cost = 0
```

```
security = SEC_NONSECURE
prop_delay = PROP_DELAY_LAN
user_def_parm_1 = 0
user_def_parm_2 = 0
user_def_parm_3 = 0
initially_active = YES
test_timeout = 5
test_retry_limit = 2
xid_timeout = 5
xid_retry_limit = 2
t1_timeout = 5
t1_retry_limit = 5

[define_ethernet_ls]
ls_name = P390HP10
description = P390 - beavs
port_name = ethl0
adj_cp_name = P390.USS3270
adj_cp_type = LEARN_NODE
mac_address = <0020af543176>
lsap_address = 0x08
auto_act_supp = NO
tg_number = 0
limited_resource = NO
solicit_sscp_sessions = NO
pu_name = <0000000000000000>
disable_remote_act = NO
default_nn_server = NO
dspu_services = NONE
dspu_name = <0000000000000000>
dlus_name = <00000000000000000000000000000000000>
bkup_dlus_name = <00000000000000000000000000000000000>
link_deact_timer = 0
use_default_tg_chars = YES
ls_attributes = SNA
adj_node_id = <00000000>
local_node_id = <00000000>
cp_cp_sess_support = YES
effect_cap = 3993600
connect_cost = 0
byte_cost = 0
security = SEC_NONSECURE
 prop_delay = PROP_DELAY_LAN
user_def_parm_1 = 0
user_def_parm_2 = 0
user_def_parm_3 = 0
target_pacing_count = 7
max_send_btu_size = 1033
ls_role = USE_PORT_DEFAULTS
```

```
initially_active = NO
react_timer = 30
react_timer_retry = 65535
test_timeout = 5
test_retry_limit = 2
xid_timeout = 5
xid_retry_limit = 2
t1_timeout = 5
t1_retry_limit = 5

[define_local_lu]
lu_name = LUHP10A
description = Test LU #1
lu_alias = LUHP10A
nau_address = 0
syncpt_support = YES
lu_session_limit = 0
default_pool = NO
pu_name = <0000000000000000>
sys_name = ""
timeout = -1
back_level = NO

[define_local_lu]
lu_name = LUHP10B
description = Test LU #2
lu_alias = LUHP10B
nau_address = 0
syncpt_support = YES
lu_session_limit = 0
default_pool = NO
pu_name = <0000000000000000>
sys_name = ""
timeout = -1
back_level = NO

[define_local_lu]
lu_name = LUHP10C
description = Test LU #3
lu_alias = LUHP10C
nau_address = 0
syncpt_support = YES
lu_session_limit = 0
default_pool = NO
pu_name = <0000000000000000>
sys_name = ""
timeout = -1
back_level = NO
```

```
[define_partner_lu]
description = APPC MVS LU for IMS
fqplu_name = P390.MVSLU01
plu_alias = MVSLU01
plu_un_name = MVSLU01
max_mc_ll_send_size = 32767
conv_security_ver = NO
parallel_sess_supp = YES

[define_partner_lu]
description = backend cics #1
fqplu_name = P390.C410XB01
plu_alias = C410XB01
plu_un_name = C410XB01
max_mc_ll_send_size = 32767
conv_security_ver = NO
parallel_sess_supp = YES

[define_partner_lu]
description = Second backend cics
fqplu_name = P390.C410XB02
plu_alias = CICS2
plu_un_name = C410XB02
max_mc_ll_send_size = 32767
conv_security_ver = NO
parallel_sess_supp = YES

[define_mode]
mode_name = SMSNA100
description = Sessions: 10 -- 5,5
max_ru_size_upp = 1024
receive_pacing_win = 4
default_ru_size = YES
max_neg_sess_lim = 256
plu_mode_session_limit = 10
min_conwin_src = 5
cos_name = #eLink
cryptography = NONE
auto_act = 0
```

# Remote Environment

A remote environment is an IBM mainframe that may or may not be on the same local network. As in the local environment, eAM software is a fully bidirectional program, supporting the remote system as either a client or server. This environment consists of the following components:

- Hardware, any workstation and network interface supported by the required software

- Software, any supported host software

## Remote Environment Configurations

Involve your mainframe system support personnel early in the process of setting up your configuration. Different individuals may be responsible for MVS, VTAM, CICS, and IMS. Make sure everyone is involved. Most of the configuration for your mainframe may have already been done.

These samples are provided for illustration. Mainframe technical support is not trivial and this information is not intended to explain all of the possible configurations. These samples represent one way a P390 can be configured to work in an APPN LAN environment.

### ATCSTRxx VTAM Start List

The following example is the VTAM start list for the BEAVS P390 machine. It supports both the subarea and APPN environments.

**Listing 2-2   VTAM Start List for BEAVS P390**

```
* -------------------------*
* VTAM START LIST FOR SYS1  *
* -------------------------*
BN=YES,
BNDYN=FULL,
XNETALS=YES,
SSCPID=06,NOPROMPT,
CONFIG=00,MAXSUBA=31,SUPP=NOSUP,
SSCPNAME=USS3270,
NETID=P390,
```

```
NODETYPE=NN,
HOSTSA=6,
CRPLBUF=(208,,15,,1,16),
IOBUF=(100,512,19,,1,20),
LFBUF=(104,,0,,1,1),
LPBUF=(64,,0,,1,1),
SFBUF=(163,,0,,1,1)
```

## XCA Major Node Defines the LAN Adapter for SYS1

This definition is set up for use with a 3172 (emulated) for connecting an APPN
network node to another APPN node. Note that it is for an Ethernet LAN, and the
SAPADDR specified must be the same as the LSap specified for the local link station.

**Listing 2-3   XCA Major Node**

```
XETH2LP1 VBUILD TYPE=XCA ** EXTERNAL COMMUNICATION ADAPT**
PORTE2   PORT  ADAPNO=1,      ** 3172 RELATIVE ADAPTER NUMBER**
               CUADDR=E22,    ** CHANNEL UNIT ADDRESS        **
               MEDIUM=CSMACD, ** LAN TYPE=ETHERNET           **
               SAPADDR=8,     ** SERVICE ACCESS POINT ADDRESS**
               TIMER=120      ** CHANNEL ACTIVATE RESP TIME  **
*
G1ETH2   GROUP DIAL=YES,      ** YES required for putype 2   **
               DYNPU=YES,
               CALL=INOUT,
               ANSWER=ON,
               ISTATUS=ACTIVE
LETH20   LINE
PETH20   PU
LETHE3   LINE
PETHE3   PU
LETHF3   LINE
PETHF3   PU
```

## Switched Network (SWNET) Definitions

The two switched network definition examples in this section (Listing 2-4 and Listing 2-5) specify the VTAM PU (Physical Unit), representing the local link stations that expect to connect with the host machine. The IDBLK and IDNUM definitions are provided to support 3270 traffic and must be unique, as well as match the values specified in the local link definition.

**Listing 2-4   SWNET Major Node (DALHP10)**

```
SWNETHHP    VBUILD   TYPE=SWNET,MAXNO=3,MAXGRP=3
P390HP10    PU  ADDR=02,
                 IDBLK=05F,
                 IDNUM=FFFFF,
                 PUTYPE=2,
                 NETID=BEALAN,
                 CPNAME=DALHP10,
                 MAXPATH=3,
                 DWACT=YES,
                 CONNTYPE=APPN,
                 CPCP=YES,
                 DYNLU=YES
* -------------------------------------------------
* SNA SAP & HP10 MAC ADDRESS BIT REVERSED FOR TRFMT
* -------------------------------------------------
PATHHP      PATH DIALNO=00041000900C24EE,
                 GRPNM=G1ETH2
LUHP10A LU  LOCADDR=0
LUHP10B LU  LOCADDR=0
LUHP10C  LU  LOCADDR=0
```

**Listing 2-5   SWNET Major Node (SUN2)**

```
SWNETH2     VBUILD   TYPE=SWNET,MAXNO=3,MAXGRP=3
P390ETH2    PU  ADDR=04,
                 IDBLK=019,
                 IDNUM=10092,
                 PUTYPE=2,
                 NETID=BEALAN,
                 CPNAME=SUN2,
                 MAXPATH=3,
                 DWACT=YES,
```

```
                    CONNTYPE=APPN,
                    CPCP=YES,
                    DYNLU=YES
* -------------------------------------------------
* SNA SAP & SUN2 MAC ADDRESS BIT REVERSED FOR TRFMT
* -------------------------------------------------
PATH01     PATH DIALNO=00081000043EE20A,
                 GRPNM=G1ETH2
LUSUN2A LU  LOCADDR=0
LUSUN2B LU  LOCADDR=0
LUSUN2C  LU  LOCADDR=0
```

## VTAM Application Program Major Node

The APPLID definition shown in Listing 2-6 defines the local stack configuration to run under OS/390 using VTAM.

**Listing 2-6   APPLID Definition (OS/390)**

```
BEASNA VBUILD TYPE=APPL
BEAAPPL1 APPL ACBNAME=BEAAPPL1,
               AUTH=(ACQ,PASS),
               APPC=YES,
               SYNCLVL=SYNCPT,
               PARSESS=YES
```

## VTAM Application Major Nodes for CICS Regions

This example (Listing 2-7) represent the partner LU definitions to be accessed from the local environment. The APPL names must match those specified in the partner LU definitions on the local machine.

**Listing 2-7   Partner LU Definitions**

```
BEACICS VBUILD TYPE=APPL              APPLICATION MAJOR NODE
* APPL DEFINITION STATEMENTS FOR CICS
* CICS 4.10 BACKEND REGION #1 SYSID=B41A
C410XB01 APPL EAS=64,                 ESTIMATED CONCURRENT SESSIONS
              MODETAB=ISTINCLM,    MAKE SURE DEFAULT MODETAB
```

```
                        PARSESS=YES,
                        AUTH=(ACQ,BLOCK,PASS) CICS CAN ACQUIRE & PASS TMLS
                                              CICS CAN REQUEST BLOCKED INPUT

C410XB02 APPL EAS=64,                 ESTIMATED CONCURRENT SESSIONS
              MODETAB=ISTINCLM,       MAKE SURE DEFAULT MODETAB
              PARSESS=YES,
              AUTH=(ACQ,BLOCK,PASS) CICS CAN ACQUIRE & PASS TMLS
                                    CICS CAN REQUEST BLOCKED INPUT
C410XB03 APPL EAS=64,                 ESTIMATED CONCURRENT SESSIONS
              MODETAB=ISTINCLM,       MAKE SURE DEFAULT MODETAB
              PARSESS=YES,
              AUTH=(ACQ,BLOCK,PASS) CICS CAN ACQUIRE & PASS TMLS
                                    CICS CAN REQUEST BLOCKED INPUT
* #######################
* END OF BEACICS APPL DEF
* #######################
```

## CICS Resource Definition Entries (RDO)

CICS connection and session definitions map the VTAM path definitions for the CICS application. Each connection represents one local LU definition in the local SNA configuration; therefore, the names must match.

CICS session definitions associate a VTAM mode with the LU specified in the connection. The mode names and session count characteristics must match those specified in the mode definitions for the local SNA configuration. Note that these definitions set AUTOCONNECT to YES, allowing automatic session acquisition for a CICS client application.

**Listing 2-8   CICS Resource Definition Entries**

```
LIST GROUP(BEAHP10) OBJECTS
GROUP NAME: BEAHP10
 ----------
 CONNECTIONS:           FHPA       FHPB       FHPC
 SESSION                           FHPA       FHPB     FHPC

CONNECTION(FHPA)        GROUP(BEAHP10)
                        DESCRIPTION(1ST HP SNAP2+ CONNECTION)
  CONNECTION-IDENTIFIERS
    NETNAME(LUHP10A)        INDSYS()
  REMOTE-ATTRIBUTES
```

```
      REMOTESYSTEM()  REMOTENAME() REMOTESYSNET()
   CONNECTION-PROPERTIES
      ACCESSMETHOD(VTAM)          PROTOCOL(APPC)    CONNTYPE()
      SINGLESESS(NO)              DATASTREAM(USER)  RECORDFORMAT(U)
      QUEUELIMIT(NO)              MAXQTIME(NO)
   OPERATIONAL-PROPERTIES
      AUTOCONNECT(NO)             INSERVICE(YES)
   SECURITY
      SECURITYNAME()              ATTACHSEC(LOCAL)  BINDSECURITY(NO)
      USEDFLTUSER(NO)
   RECOVERY
      PSRECOVERY(SYSDEFAULT)

CONNECTION(FHPB)       GROUP(BEAHP10)
                       DESCRIPTION(2ND HP SNA+ 2 CONNECTION)
   CONNECTION-IDENTIFIERS
      NETNAME(LUHP10B)            INDSYS()
   REMOTE-ATTRIBUTES
      REMOTESYSTEM()              REMOTENAME()      REMOTESYSNET()
   CONNECTION-PROPERTIES
      ACCESSMETHOD(VTAM)          PROTOCOL(APPC)    CONNTYPE()
      SINGLESESS(NO)              DATASTREAM(USER)  RECORDFORMAT(U)
      QUEUELIMIT(NO)              MAXQTIME(NO)
   OPERATIONAL-PROPERTIES
      AUTOCONNECT(NO)             INSERVICE(YES)
   SECURITY
      SECURITYNAME()              ATTACHSEC(LOCAL)  BINDSECURITY(NO)
      USEDFLTUSER(NO)
   RECOVERY
      PSRECOVERY(SYSDEFAULT)

CONNECTION(FHPC)       GROUP(BEAHP10)
                       DESCRIPTION(3RD HP SNA+ 2 CONNECTION)
    CONNECTION-IDENTIFIERS
       NETNAME(LUHP10C)           INDSYS()
    REMOTE-ATTRIBUTES
       REMOTESYSTEM()             REMOTENAME()      REMOTESYSNET()
    CONNECTION-PROPERTIES
       ACCESSMETHOD(VTAM)         PROTOCOL(APPC)    CONNTYPE()
       SINGLESESS(NO)             DATASTREAM(USER)  RECORDFORMAT(U)
       QUEUELIMIT(NO)             MAXQTIME(NO)
    OPERATIONAL-PROPERTIES
       AUTOCONNECT(NO)            INSERVICE(YES)
    SECURITY
       SECURITYNAME()             ATTACHSEC(LOCAL)  BINDSECURITY(NO)
       USEDFLTUSER(NO)
    RECOVERY
       PSRECOVERY(SYSDEFAULT)
```

```
SESSIONS(FHPA)        GROUP(BEAHP10)
                      DESCRIPTION(1ST HP SNAP2+ SESSION)
   SESSION-IDENTIFIERS
     CONNECTION(FHPA)         SESSNAME()         NETNAMEQ()
     MODENAME(SMSNA100)
   SESSION-PROPERTIES
     PROTOCOL(APPC)           MAXIMUM(32,16)     RECEIVEPFX()
     RECEIVECOUNT()           SENDPFX()          SENDCOUNT()
      SENDSIZE(4096)          RECEIVESIZE(4096) SESSPRIORITY(0)
   PRESET-SECURITY
     USERID()
   OPERATIONAL-PROPERTIES
     AUTOCONNECT(YES)         BUILDCHAIN(YES)    USERAREALEN(0)
     IOAREALEN(0,0)           RELREQ(NO)         DISCREQ(NO)
     NEPCLASS(0)
   RECOVERY
     RECOVOPTION(SYSDEFAULT)

SESSIONS(FHPB)        GROUP(BEAHP10)
                      DESCRIPTION(2ND HP SNAP2+ SESSION)
   SESSION-IDENTIFIERS
     CONNECTION(FHPB)         SESSNAME()         NETNAMEQ()
     MODENAME(SMSNA100)
   SESSION-PROPERTIES
     PROTOCOL(APPC)           MAXIMUM(32,16)     RECEIVEPFX()
     RECEIVECOUNT()           SENDPFX()          SENDCOUNT()
     SENDSIZE(4096)           RECEIVESIZE(4096) SESSPRIORITY(0)
   PRESET-SECURITY
     USERID()
   OPERATIONAL-PROPERTIES
     AUTOCONNECT(YES)         BUILDCHAIN(YES)    USERAREALEN(0)
     IOAREALEN(0,0)           RELREQ(NO)         DISCREQ(NO)
     NEPCLASS(0)
   RECOVERY
     RECOVOPTION(SYSDEFAULT)

SESSIONS(FHPC)        GROUP(BEAHP10)
                      DESCRIPTION(3RD HPSNAP2+ SESSION)
   SESSION-IDENTIFIERS
     CONNECTION(FHPC)         SESSNAME()         NETNAMEQ()
     MODENAME(SMSNA100)
   SESSION-PROPERTIES
     PROTOCOL(APPC)           MAXIMUM(10,5)      RECEIVEPFX()
     RECEIVECOUNT()           SENDPFX()          SENDCOUNT()
     SENDSIZE(4096)           RECEIVESIZE(4096) SESSPRIORITY(0)
   PRESET-SECURITY
     USERID()
   OPERATIONAL-PROPERTIES
     AUTOCONNECT(YES)         BUILDCHAIN(YES)    USERAREALEN(0)
```

```
        IOAREALEN(0,0)              RELREQ(NO)              DISCREQ(NO)
        NEPCLASS(0)
RECOVERY
  RECOVOPTION(SYSDEFAULT)
```

# Sample JCL for an MVS Platform

Job Control Language (JCL) is used on an OS/390 Multiple Virtual Storage (MVS) platform to set the environment and invoke CRM commands. The following sections provide samples of JCL that may be used for your MVS operating system. The sample SET commands may not reflect the configuration of your system. You must customize the SET commands for your environment. Refer to your System Administrator for more information about your particular setup. Refer to Appendix A, "Command Reference Pages," for more information about the SNACRM, CRMLKON, CRMLKOFF, CRMDOWN, and CRMLOGS commands.

## Sample JCL for the SNACRM Command

Following is an explanation of the SET commands and an example of JCL that can be used when you run the SNACRM command.

SET SNACMD — Sets the SNACRM command line parameters. Refer to SNACRM in Appendix A, "Command Reference Pages" for more information about the command line parameters.

SET OBJLIB — Indicates the name of the PDSE library where the SNACRM executable is installed.

SET DATA1 — Indicates the name of the PDS library where the SNACRM required parameter file FMB was installed.

SET DATA2 — Indicates the dataset containing the ENVFILE.

SET ENVFILE — Indicates the name of the PDS member that contains the environment variables for the SNACRM. A sample member, ENV, is delivered with your product.

SET SIZE     Defines the region size for the running SNACRM task. The recommended setting for this option is 0M to allow the SNACRM to start up and level out to the size it requires.

SET ENV     Indicates the ENVFILE DD name.

SET CEE     Specifies the high-level qualifier for the LE runtime library. CEE should be set to the prefix of the IBM Language Environment data sets. Language Environment is required to run the SNACRM.

SET CBC     Specifies the high-level qualifier for the C/C++ runtime library. CBC should be set to the prefix of the IBM C++ data sets.

**Listing 2-9   Sample JCL for SNACRM Command**

```
//*************************************************************
**
//*  USE THE SET STATEMENTS TO SET THE APPROPRIATE VALUES      *
//*   SNACMD IS THE SNACRM COMMAND LINE                        *
//*   DATA1 IS THE DATA SET CONTAINING FMB                     *
//*   DATA2 IS THE DATA SET CONTAINING ENVFILE                 *
//*  ENVFILE IS THE FILE CONTAINING ENVIRONMENT VARIABLES      *
//*   ENV SETS THE ENVFILE DD NAME                             *
//*   RUNOPTS SETS ANY DESIRED LE RUNTIME OPTIONS              *
//*   SIZE SETS THE REGION SIZE FOR THE SNACRM PROCESS         *
//*  OBJLIB IS THE LOAD LIBRARY CONTAINING THE eAM PROGRAM OBJECTS*
//*   CEE IS THE HLQ FOR THE LE RUNTIME LIBRARY                *
//*   CBC IS THE HLQ FOR THE C/C++ RUNTIME LIBRARY             *
//*************************************************************
**
//  SET SNACMD='"//<address>:<port>" <group>'
//   SET OBJLIB=
//   SET DATA1=
//   SET DATA2=
//   SET ENVFILE=ENV
//   SET SIZE=0M
//  SET ENV='ENVAR("_CEE_ENVFILE=DD:ENV")'
//   SET CEE=CEE,CBC=CBC
//  SNACRM  EXEC PGM=SNACRM,REGION=&SIZE,
//  PARM='POSIX(ON) &ENV &RUNOPTS/&SNACMD'
//  STEPLIB DD DSN=&OBJLIB,DISP=SHR
//       DD  DSN=&CEE..SCEERUN,DISP=SHR
//       DD  DSN=&CBC..SCLBDLL,DISP=SHR
//FLDTBL   DD DSN=&DATA1,DISP=SHR
//ENV     DD DSN=&DATA2(&ENVFILE),DISP=SHR
//
```

# Sample JCL for the CRMLKON Command

The following section is an explanation of the SET commands and an example of JCL that can be used when you run the CRMLKON command.

SET LNKCMD    Sets the CRMLKON command line parameters. Refer to crmlkon in Appendix A, "Command Reference Pages" for more information about the command line parameters.

SET OBJLIB    Indicates the name of the PDSE library where the CRMLKON executable is installed.

SET DATA1    Indicates the name of the PDS library where the CRMLKON required parameter file FMB was installed.

SET DATA2    Indicates the dataset containing the ENVFILE.

SET ENVFILE    Indicates the name of the PDS member that contains the environment variables for the CRMLKON. A sample member, ENV, is delivered with your product.

SET SIZE    Defines the region size for the running CRMLKON task.

SET ENV    Indicates the ENVFILE DD name.

SET CEE    Specifies the high-level qualifier for the LE runtime library. CEE should be set to the prefix of the IBM Language Environment data sets. Language Environment is required to run CRMLKON.

SET CBC    Specifies the high-level qualifier for the C/C++ runtime library. CBC should be set to the prefix of the IBM C++ data sets.

**Listing 2-10  Sample JCL for CRMLKON Command**

```
//**************************************************************
**
//*   LINKCMD INDICATES THE DISTRIBUTED SNACRM ADDRESS AND LINKNAME*
//*   RUNOPTS SETS ANY DESIRED LE RUNTIME OPTIONS               *
//*   SIZE SETS THE REGION SIZE FOR THE SNACRM PROCESS          *
//*   OBJLIB IS THE LOAD LIBRARY CONTAINING THE eAM PROGRAM OBJECTS*
//*   CEE IS THE HLQ FOR THE LE RUNTIME LIBRARY                 *
//*   CBC IS THE HLQ FOR THE C/C++ RUNTIME LIBRARY              *
```

```
//*************************************************************
**
//   SET LINKCMD='-n<host name>:<port><linkname>'
//   SET OBJLIB=
//   SET RUNOPTS=
//   SET DATA1=
//   SET DATA2=
//   SET SIZE=1M
//   SET ENV='ENVAR("_CEE_ENVFILE=DD:ENV")'
//   SET CEE=CEE,CBC=CBC
//CRMLKON  EXEC  PGM=CRMLKON,REGION=&SIZE,
//   PARM='POSIX(ON) &ENV &RUNOPTS/&LINKCMD'
//STEPLIB  DD  DSN=&OBJLIB,DISP=SHR
//         DD  DSN=&CEE..SCEERUN,DISP=SHR
//         DD  DSN=&CBC..SCLBDLL,DISP=SHR
//FLDTBL   DD  DSN=&DATA1,DISP=SHR
```

# Sample JCL for the CRMLKOFF Command

The following section is an explanation of the SET commands and an example of JCL that can be used when you run the CRMLKOFF command.

SET LNKCMD
Sets the CRMLKOFF command line parameters. Refer to crmlkoff in Appendix A, "Command Reference Pages" for more information about the command line parameters.

SET OBJLIB
Indicates the name of the PDSE library where the CRMLKOFF executable is installed.

SET DATA1
Indicates the name of the PDS library where the CRMLKOFF required parameter file FMB was installed.

SET DATA2
Indicates the dataset containing the ENVFILE.

SET ENVFILE
Indicates the name of the PDS member that contains the environment variables for the CRMLKOFF. A sample member, ENV, is delivered with your product.

SET SIZE
Defines the region size for the running CRMLKOFF task.

SET ENV
Indicates the ENVFILE DD name.

SET CEE
Specifies the high-level qualifier for the LE runtime library. CEE should be set to the prefix of the IBM Language Environment data sets. Language Environment is required to run CRMLKOFF.

SET CBC    Specifies the high-level qualifier for the C/C++ runtime library. CBC should be set to the prefix of the IBM C++ data sets.

**Listing 2-11   Sample JCL for CRMLKOFF Command**

```
//***************************************************************
**
//*  LINKCMD INDICATES THE DISTRIBUTED SNACRM ADDRESS AND LINKNAME*
//*  RUNOPTS SETS ANY DESIRED LE RUNTIME OPTIONS               *
//*  SIZE SETS THE REGION SIZE FOR THE SNACRM PROCESS          *
//*  OBJLIB IS THE LOAD LIBRARY CONTAINING THE eAM PROGRAM OBJECTS*
//*  CEE IS THE HLQ FOR THE LE RUNTIME LIBRARY                 *
//*  CBC IS THE HLQ FOR THE C/C++ RUNTIME LIBRARY              *
//***************************************************************
**
//   SET LINKCMD='-n<host name>:<port><linkname>'
//   SET OBJLIB=
//   SET RUNOPTS=
//   SET DATA1=
//   SET DATA2=
//   SET SIZE=1M
//   SET ENV='ENVAR("_CEE_ENVFILE=DD:ENV")'
//   SET CEE=CEE,CBC=CBC
//CRMLKOFF  EXEC  PGM=CRMLKOFF,REGION=&SIZE,
//   PARM='POSIX(ON) &ENV &RUNOPTS/&LINKCMD'
//STEPLIB  DD  DSN=&OBJLIB,DISP=SHR
//         DD  DSN=&CEE..SCEERUN,DISP=SHR
//         DD  DSN=&CBC..SCLBDLL,DISP=SHR
//FLDTBL   DD  DSN=&DATA1,DISP=SHR
```

# Sample JCL for the `CRMDOWN` Command

The following section is an explanation of the SET commands and an example of JCL that can be used when you run the CRMDOWN command.

SET LNKCMD    Sets the CRMDOWN command line parameters. Refer to CRMDOWN in Appendix A, "Command Reference Pages" for more information about the command line parameters.

SET OBJLIB    Indicates the name of the PDSE library where the CRMDOWN executable is installed.

SET DATA1    Indicates the name of the PDS library where the CRMDOWN required parameter file FMB was installed.

SET DATA2    Indicates the dataset containing the ENVFILE.

SET ENVFILE   Indicates the name of the PDS member that contains the environment variables for the CRMDOWN. A sample member, ENV, is delivered with your product.

SET SIZE    Defines the region size for the running CRMDOWN task.

SET ENV    Indicates the ENVFILE DD name.

SET CEE    Specifies the high-level qualifier for the LE runtime library. CEE should be set to the prefix of the IBM Language Environment data sets. Language Environment is required to run CRMDOWN.

SET CBC    Specifies the high-level qualifier for the C/C++ runtime library. CBC should be set to the prefix of the IBM C++ data sets.

**Listing 2-12   Sample JCL for CRMDOWN Command**

```
//****************************************************************
**
//*  LINKCMD INDICATES THE DISTRIBUTED SNACRM ADDRESS           *
//*  RUNOPTS SETS ANY DESIRED LE RUNTIME OPTIONS                *
//*  SIZE SETS THE REGION SIZE FOR THE SNACRM PROCESS           *
//*  OBJLIB IS THE LOAD LIBRARY CONTAINING THE eAM PROGRAM OBJECTS*
//*  CEE IS THE HLQ FOR THE LE RUNTIME LIBRARY                  *
//*  CBC IS THE HLQ FOR THE C/C++ RUNTIME LIBRARY              *
//****************************************************************
**
//   SET LINKCMD='-n<host name>:<port> '
//   SET OBJLIB=
//   SET RUNOPTS=
//   SET DATA1=
//   SET DATA2=
//   SET SIZE=1M
//   SET ENV='ENVAR("_CEE_ENVFILE=DD:ENV")'
//   SET CEE=CEE,CBC=CBC
//CRMDOWN  EXEC  PGM=CRMDOWN,REGION=&SIZE,
// PARM='POSIX(ON) &ENV &RUNOPTS/&LINKCMD'
//STEPLIB  DD  DSN=&OBJLIB,DISP=SHR
//         DD  DSN=&CEE..SCEERUN,DISP=SHR
//         DD  DSN=&CBC..SCLBDLL,DISP=SHR
//FLDTBL   DD  DSN=&DATA1,DISP=SHR
```

# Sample JCL for the CRMLOGS Command

The following section is an explanation of the SET commands and an example of JCL that can be used when you run the CRMLOGS command.

SET LNKCMD    Sets the CRMLOGS command line parameters. Refer to CRMLOGS in Appendix A, "Command Reference Pages" for more information about the command line parameters.

SET OBJLIB    Indicates the name of the PDSE library where the CRMLOGS executable is installed.

SET DATA1     Indicates the name of the PDS library where the CRMLOGS required parameter file FMB was installed.

SET DATA2     Indicates the dataset containing the ENVFILE.

SET ENVFILE   Indicates the name of the PDS member that contains the environment variables for the CRMLOGS. A sample member, ENV, is delivered with your product.

SET SIZE      Defines the region size for the running CRMLOGS task.

SET ENV       Indicates the ENVFILE DD name.

SET CEE       Specifies the high-level qualifier for the LE runtime library. CEE should be set to the prefix of the IBM Language Environment data sets. Language Environment is required to run CRMLOGS.

SET CBC       Specifies the high-level qualifier for the C/C++ runtime library. CBC should be set to the prefix of the IBM C++ data sets.

**Listing 2-13   Sample JCL for CRMLOGS Command**

```
//************************************************************
**
//*  THIS JOB IS USED TO CHECK THE RECOVERY LOGS FOR         *
//*  OUTSTANDING TRANSACTION DATA. SEE USER GUIDE FOR MORE INFO  *
//*                                                          *
//* @(#)$Id: crmlogs.jcl,v 1.3 2000/05/17 16:20:27 crount Exp $
//* Copyright (c)2000 BEA Systems, Inc., all rights reserved.    *
//************************************************************
```

```
**
//*  SNAGRP IS THE GROUP NAME OF THE DISTRIBUTED SNACRM        *
//*  DATA1 IS THE DATA SET CONTAINING FMBDEF                   *
//*  DATA2 IS THE DATA SET CONTAINING ENVFILE                  *
//*  ENVFILE IS THE FILE CONTAINING ENVIRONMENT VARIABLES      *
//*  ENV SETS THE ENVFILE DD NAME                              *
//*  RUNOPTS SETS ANY DESIRED LE RUNTIME OPTIONS               *
//*  SIZE SETS THE REGION SIZE FOR THE SNACRM PROCESS          *
//*  OBJLIB IS THE LOAD LIBRARY CONTAINING THE EAM PROGRAM OBJECTS*
//*  CEE IS THE HLQ FOR THE LE RUNTIME LIBRARY                 *
//*  CBC IS THE HLQ FOR THE C/C++ RUNTIME LIBRARY              *
//*************************************************************
**
//  SET SNACMD=<group>
//  SET OBJLIB=
//  SET DATA1=
//  SET DATA2=
//  SET RUNOPTS=
//  SET SIZE=10M
//  SET ENV='ENVAR("_CEE_ENVFILE=DD:ENV")'
//  SET CEE=CEE,CBC=CBC
//CRMLOGS  EXEC  PGM=CRMLOGS,REGION=&SIZE,
//  PARM='POSIX(ON) &ENV &RUNOPTS/&SNAGRP'
//STEPLIB  DD  DSN=&OBJLIB,DISP=SHR
//         DD  DSN=&CEE..SCEERUN,DISP=SHR
//         DD  DSN=&CBC..SCLBDLL,DISP=SHR
//MSGFILE  DD  SYSOUT=*
//SYSPRINT DD  SYSOUT=*
//FLDTBL   DD  DSN=&DATA1,DISP=SHR
//ENV      DD  DSN=&DATA2(ENV),DISP=SHR
//
```

# 3 Administering the SNA Components

The topics in this section cover activities an administrator performs with the Communications Resource Manager (CRM) to maintain BEA eLink Adapter for Mainframe (eAM) applications.

The interface to the stack administration and configuration is dependent on the stack provider and thus cannot be covered in this guide. Refer to vendor publications for the stack(s) used in your environment.

This section discusses the following topics:

- Starting the CRM on UNIX or Windows NT

- Starting the CRM on OS/390 MVS

- Using the CRM Monitor

- Setting CRM and Stack Traces

- Activating and De-Activating Links

- Stopping the CRM on OS/390

# Starting the CRM on UNIX or Windows NT

The CRM is a server that communicates directly with the PU 2.1 server to provide SNA connectivity. These servers can be started manually. The PU 2.1 server must always be started before the CRM. Both servers must be started before starting the associated eAM gateway (GWSNAX).

To ensure proper start up of the CRM, complete the following tasks:

1. Set the following environment variables in the environment where the CRM is started:

   FIEDLTBLS32 must contain `fmb.def`

   FLDTBLDIR32 must contain the path to the eAM libraries directory

   APPDIR must be set to the application directory

2. Start the PU2.1 Server.

   Refer to the operational documentation provided by your SNA stack vendor for information about starting the PU2.1 server. The SNA stack must be running and active before you start the CRM.

3. Start the CRM.

   - To start the CRM on UNIX:

     Enter the SNACRM command on the command line.

     Refer to Appendix A, "Command Reference Pages" for more detailed information about the SNACRM command.

     When you start the CRM from the UNIX command line, the CRM Command Line Console puts its prompt in a window, and if exited, shuts down all of the active links.

   - To start the CRM from a Command Prompt Window on Windows NT:

   a. Open a Command Prompt window.

   b. Navigate to the drive where the CRM executable is installed.

   c. Enter SNACRM to start SNACRM.exe.

On Windows NT, the CRM can be started in its own window; however, no prompt is displayed and no console commands are available. Refer to "Command Reference Pages," for more detailed information about the `SNACRM` command.

# Starting the CRM on OS/390 MVS

The OS/390 MVS platform sets the environment and invokes the CRM through Job Control Language (JCL).

1. Set the following environment variables in the environment where the CRM is started. A sample file is delivered (ENV) in the data library.

   ```
   FLDTBLDIR32=DD:FLDTBL
   FIELDTBLS32=FMB
   APPDIR=<High level qualifier for datasets to be created in
   APPDIR>
   ```

2. Run a `SNACRM` job using JCL written for your system.

   Refer to "Sample JCL for the SNACRM Command" and "Command Reference Pages," for more information about sample `SNACRM` JCL and the `SNACRM` command.

**Note:** If the CRM is installed on an OS/390 MVS platform, it does not have to be restarted if the eAM gateway goes down abnormally. If a `tmshutdown` command is issued, the CRM will shut down.

Only the OS/390 MVS version and the OS/390 UNIX version of the CRM have this persistent feature.

# Using the CRM Monitor

You can use the CRM monitor to set trace levels for a selected CRM and the associated APPC stacks. You also can observe link activity and display trace status, link status, and link statistics.

> **Note:** The CRM monitor does not show trace data. This data is captured in a file under the APPDIR directory (where APPDIR is the variable name associated with the application directory). Please contact BEA Customer Support for help in locating the trace files and interpreting them.

The BEA eLink for Mainframe software includes two utilities that launch and execute the SNACRM monitor. The xsnacrm utility is designed for UNIX platforms and requires Motif libraries. The jsnacrm utility is designed for Windows NT platforms and supplies both a Java-based application and an applet.

> **Note:** The jsnacrm utility can only be run from an Windows NT platform. It must be run on a UNIX or Windows NT platform to connect to an OS/390 CRM. The xsnacrm utility can only be run from the UNIX and Windows NT platform.

# Using the CRM Monitor and the xsnacrm Utility

The xsnacrm utility provides realtime monitoring of the CRM and displays information describing the activity occurring in each CRM. The xsnacrm utility should be used by administrators and system operators only. Access to this utility may be restricted by setting the execute permissions during installation. The xsnacrm utility requires Motif libraries.

To launch and execute the CRM monitor using the xsnacrm utility, perform the following steps:

1. Determine and enter the appropriate trace level:

   - *0*—No tracing. Setting this level effectively disables SNACRM tracing and closes the trace file if there is one. If tracing is subsequently restarted, a new file will be created with an incremented numerical suffix.

   - *1*—Minimum tracing. At this level, the SNACRM traces only major events and is sufficient only to determine the sequence of application conversations.

   - *2*—Medium tracing. At this level, SNACRM also traces all I/O buffers.

   - *3*—Maximum tracing. At this level, SNACRM also traces all APPC verbs.

   The APPC Protocol Stack API trace is either **enabled** or **disabled**. If enabled, it generally shows the parameters and results of all API calls. Depending on the

Stack being used, other options (such as vendor-specified environment variables) may have to be activated for SNACRM to enable the trace.

2. Use `address` to specify the host name and port number of a CRM to monitor. This value must match the corresponding parameter on the command line used to start the CRM you wish to monitor.

   At least one address must be specified. Multiple CRMs may be monitored by specifying all their associated addresses.

   The format of an address consists of two forward slashes (//) followed by a host name or address, a colon (:) , and a service name or decimal port number (containing no spaces).

   If a host name is used, it should be an entry in the file `/etc/hosts`. If a host address is used, it should be specified in the format `nnn.nnn.nnn.nnn` where each group of `nnn` represents a decimal number between 1 and 255. This host should identify the computer where the CRM you wish to monitor is running, not the host where `xsnacrm` is to run.

   If a service name is used, it should be an entry in the file `/etc/services`. If a decimal port number is used, it should be a decimal number in the range 4000 - 32767. This number must match the corresponding port number on the command line used to start the CRM you want to monitor. (If the CRM was started automatically, the address is specified in the `DMCONFIG` file).

3. Customize `X Resources`. The default `X Resources` for `xsnacrm` correspond to the distributed contents of the associated file `xsnacrm`. To customize the application, copy the `xsnacrm` file to your home directory and edit it.

4. Start the `xsnacrm` utility by entering the following command:

   ```
   xsnacrm [<X/Motif args>] <addr> [ <addr> ...]
   ```

   In the above command, the following definitions apply:

   <addr>

   > The INET address of a CRM to monitor  and  `<addr>` is specified as either `0xHHHHHHHHHHHHHHHH` or `//<host>:<port>`

   <host>

   > Either a host name or `an INET host address (e.g. 204.29.36.44) and`

A service name or a decimal port number.

Figure 3-1 illustrates the xsnacrm screen.

**Figure 3-1   xsnacrm Screen**
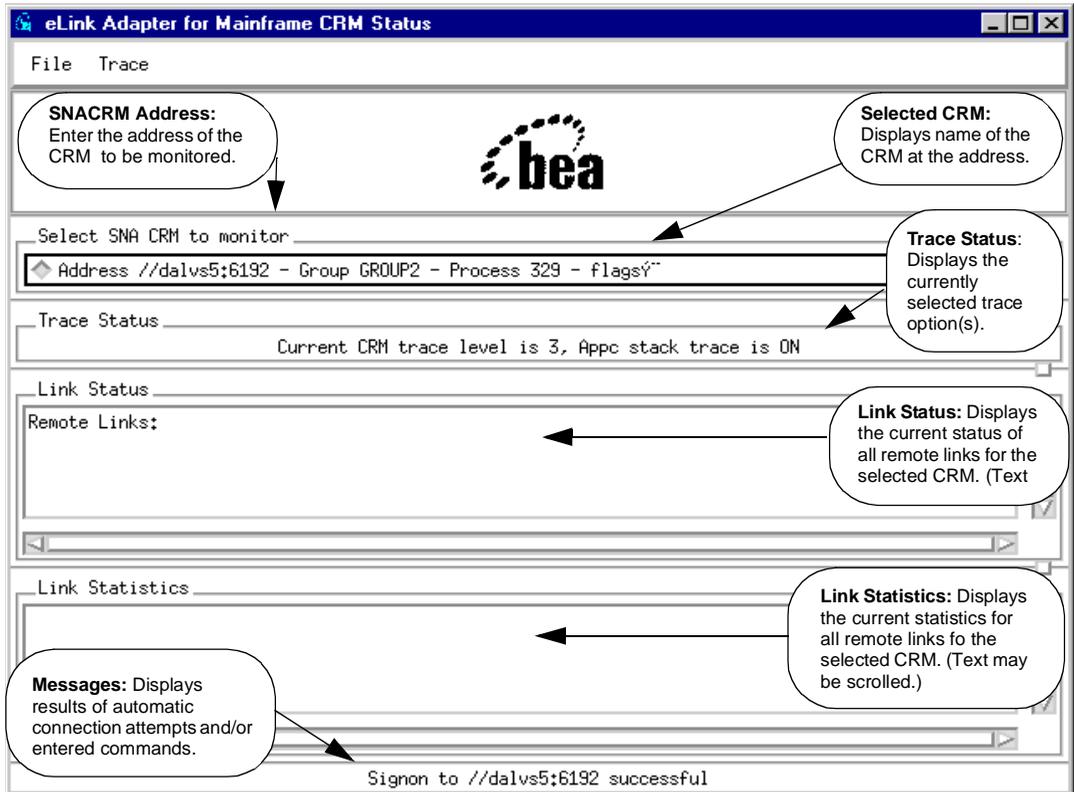


# Using the CRM Monitor and the jsnacrm Utility

The jsnacrm utility is written in Java as both an application and an applet. The application launches and executes like any other Java application and can be set up so it is accessible from the Windows desktop. The applet launches and executes from a network browser.

The BEA eLink Java Adapter for Mainframe software CD-ROM contains the following files associated with the `jsnacrm` utility:

- `bealogo.gif`
- `jsnacrm.html`
- `jsnacrm.jar`
- `moncrm.jar`
- `moncrm.x509`

## Running the Java Applet Version

The following sections describe how to set up and run the Java applet version of the `jsnacrm` utility.

## Prerequisites for Running the Java Applet Version

To run the Java applet version, you must complete the following tasks:

1. Verify that either Netscape Communicator 4.x or Internet Explorer 4.x or 5.x is installed on your Windows NT system.

2. Verify that the Java plug-in is installed on your system. You can download this plug-in from the following internet location:

   `http://java.sun.com/products/plugin`

   **Note:** If the Java plug-in is not already installed on your system when you attempt to open the `jsnacrm.html` file, the program prompts you for an automatic download of the plug-in by the browser.

3. Set your system to accept code signed by the identity, `moncrm`. To do this, perform the following steps:

   d. Create the `moncrm` identity in your JDK 1.2.2 identity database. Enter the parameter `true` to establish `moncrm` as a trusted identity.

   `javakey -c moncrm true`

   e. Import the `moncrm` certificate into your identity database. To associate the certificate with the identity, use the nickname `moncrm` as the first argument to the `javakey` command.

```
javakey -ic moncrm %TUXDIR%\bin\moncrm.x509
```

## Starting the Java Applet

To start the Java applet in an existing browser, open the file:

```
<tuxedo-path>\bin\jsnacrm.html
```

To build a shortcut to start the Java applet using a separate instance of your network browser, enter the following command:

```
<browser-pathname> %TUXDIR%\bin\jsnacrm.html
```

## Running the Java Applet Version

To run the Java applet version, perform the following tasks:

1. Set your applet version to monitor either a local or remote CRM by making selections on the Java Plug-in Properties control panel. This control panel is automatically downloaded with the plug-in and is initiated from the Windows Start Programs pop-up menu. Refer to online documentation about the control panel at the following Internet location:

   ```
   http://java.sun.com/products/plugin/1.1.1/docs
   ```

2. Enter the address of the CRM that you want to monitor in the field at the top of the screen when the Monitor screen displays (See Figure 3-2).

   a. To monitor a local CRM, select Applet Host from the Network Access drop-down menu. Type the following address in the **Enter CRM Address** panel:

      ```
      //localhost:port
      ```

      The following definitions apply to this address:

      `localhost` explicitly specifies the local host.

      `port` specifies the port number of the CRM on the local host.

   b. To monitor a remote CRM, select **Unrestricted** from the **Network Access** drop down menu. Type the following address in the **Enter CRM Address** panel:

      ```
      //remotehostname:address
      ```

      The following definitions apply to this address:

remotehostname specifies the remote host.

address specifies the network address of the CRM on the remote host.

The CRM Monitor contains two screen areas that require user entry and four screen areas that display information about the CRM being monitored. Status messages are displayed at the bottom of the screen. The screen functions are listed in Table 3-1 and shown in Figure 3-2.

**Table 3-1  CRM Monitor Screen Functions**

| Display Section | Function |
|---|---|
| CRM Address | Enter the address of the CRM to be monitored. |
| Selected CRM | Displays the name of the CRM at the address entered in the address field. |
| Trace Status | Displays the currently selected trace options. |
| Trace Options | ■ **Stop CRM Trace** disables CRM tracing and closes the trace file, if it exists.<br>■ **Set Minimum CRM Trace** establishes tracing of only major events. This level is sufficient only to determine the sequence of application conversations.<br>■ **Set Medium CRM Trace** establishes minimum tracing plus tracing of all I/O buffers.<br>■ **Set Maximum CRM Trace** establishes medium tracing plus tracing of all APPC verbs.<br>■ **Start APPC Stack Trace** establishes tracing of the APPC stack. Generally shows the parameters and results of all API calls. Depending on the stack being used, other options such as vendor-specified environment variables also may have to be activated. May be selected along with any of the previous Trace Options.<br>■ **Stop APPC Stack Trace** disables APPC stack trace, if established. |
| Link Status | Displays the current status of all remote links for the selected CRM. (Text may be scrolled.) |
| Link Statistics | Displays the current statistics for all remote links for the selected CRM. (Text may be scrolled.) |

**Table 3-1  CRM Monitor Screen Functions**

| Display Section | Function |
|---|---|
| Message Line | Displays messages showing either the results of automatic connection attempts or commands issued to change the trace options. |

**Figure 3-2   The CRM Monitor Running as an Applet on a Network Browser**
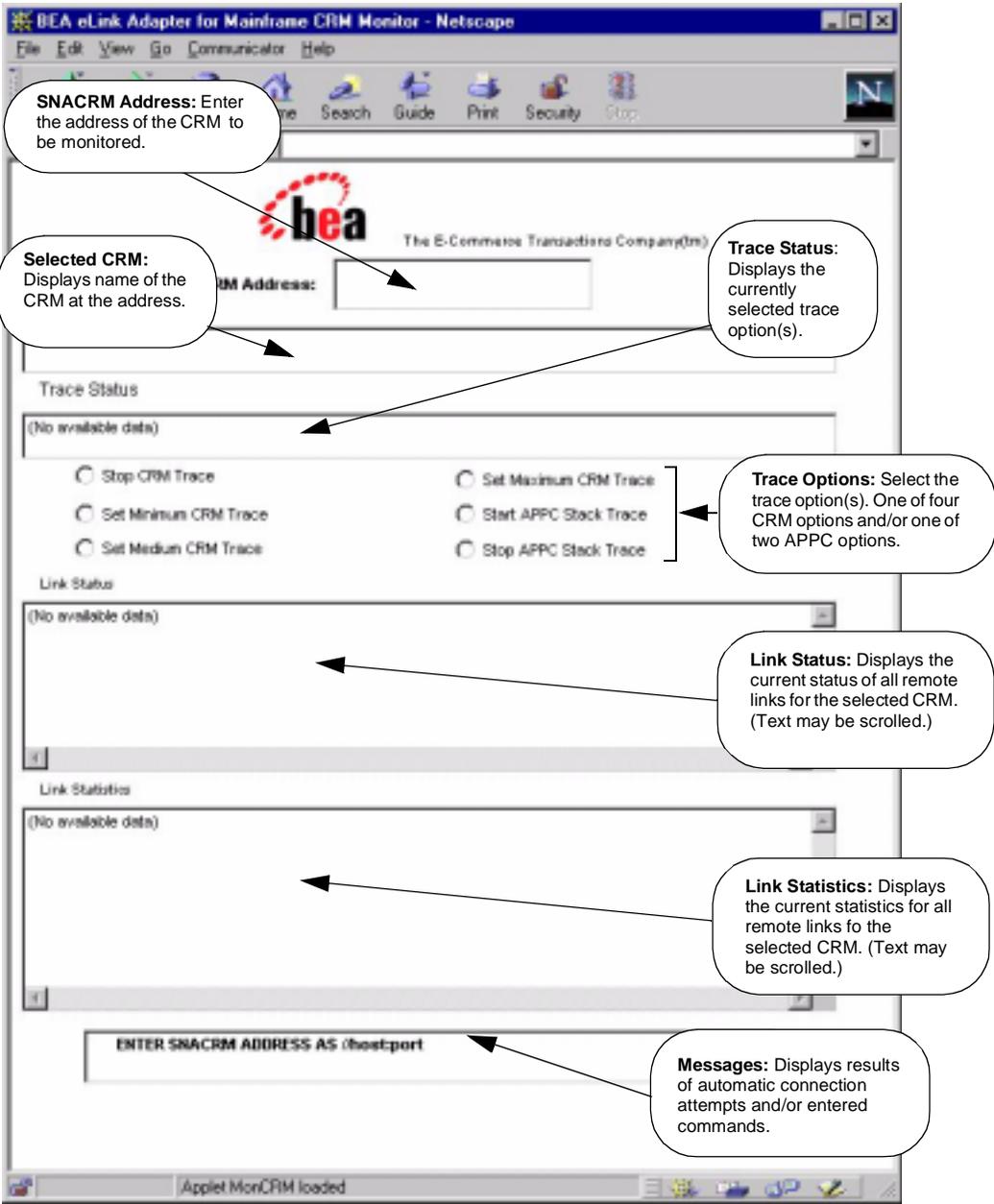
# Running the Java Application Version

The Java application version displays and operates identically to the applet version. Refer to screen definitions and functions discussed under "Running the Java Applet Version."

To build a shortcut for starting the Java application version, perform the following steps:

1. Enter the command:

   ```
   jrew -classpath %ClassPath%;jsnacrm.jar jsnacrm
   ```

2. Start the application in the directory `%TUXDIR%\bin` so it can find its files.

To run from a command window, perform the following steps:

1. Change directory to `%TUXDIR%\bin`.

2. Enter the command:

   ```
   jrew -classpath %ClassPath%;jsnacrm.jar jsnacrm
   ```

# Setting CRM and Stack Traces

Trace capabilities are provided for the CRM process. To initiate stack tracing, include the -t option for the SNACRM command with a number to specify the level of tracing. Refer to "Trace Options" for the SNACRM command in Appendix A, "Command Reference Pages."

The APPC Stack API trace is either enabled or disabled using the -s option of the SNACRM command. Refer to "General Options" for the SNACRM command in Appendix A, "Command Reference Pages." When enabled, the trace generally shows the parameters and results of all API calls. Depending on the stack being used, other options (such as vendor-specified environment variables) may have to be activated for SNACRM to enable the trace. Consult the vendor publications for instructions on how to set up stack tracing.

**Note:** On OS/390 platforms, the General Trace Facility (GFT) is used to capture API trace records under user EID 2EA. The GTF must be active on the OS/390 platform to use the `-s` parameter.

# Activating and De-Activating Links

The eAM gateway provides a command line tool  to activate and de-activate links that have been defined in the `DM_SNALINKS` section of the `dmconfig` file. This tool consists of two commands and their associated parameters:

■ `crmlkon`

Start one or more SNA links with this command.

● To activate the SNA links on UNIX or Windows NT, use the following syntax:

`crmlkon -n<hostname:port> [-v -i -h] <linkname> ...`

Refer to `crmlkon` in Appendix A, "Command Reference Pages" for more information on the `crmlkon` command.

● To activate the SNA links on OS/390 MVS:

Run a `CRMLKON` job using JCL written explicitly for your system. Refer to "Sample JCL for CRMLKON Command" and Appendix A, "Command Reference Pages" for more information about sample `CRMLKON` JCL and the `crmlkon` command.

**Note:** You will not receive notification that the link(s) started with the `crmlkon` command are activated. Use the CRM monitor to verify a link is active. Refer to "Using the CRM Monitor" for more information.

■ `crmlkoff`

Stop one or more SNA links with this command.

● To de-activate the SNA links on UNIX or Windows NT, use the following syntax:

`crmlkoff -n<hostname:port> [-v -i -h] <linkname> ...`

Refer to crmlkoff in Appendix A, "Command Reference Pages" for more information on the crmlkoff command.

● To de-activate the SNA links on OS/390 MVS:

Run a CRMLKOFF job using JCL written explicitly for your system. Refer to "Sample JCL for CRMLKOFF Command" and Appendix A, "Command Reference Pages" for more information about sample CRMLKOFF JCL and the crmlkoff command.

**Note:** You will not receive notification that the links stopped with the crmlkoff command are de-activated. Use the CRM monitor to verify that a link is not active. Refer to "Using the CRM Monitor" for more information.

If a link to a remote host is deactivated and reactivated by the host, the eAM software normally re-establishes the link automatically. If this does not occur, use the crmlkon command to re-establish the link.

# Stopping the CRM on OS/390

If the eAM gateway terminates abnormally, the persistent CRM feature allows the OS/390 CRM to continue running. (A tmshutdown command is considered a normal termination.) You must explicitly shutdown the CRM if the eAM gateway has terminated abnormally and the CRM needs to be stopped.

■ To stop the CRM from a UNIX or Windows NT system:

Enter the crmdown command on the command line. Refer to crmdown in Appendix A, "Command Reference Pages" for more detailed information about the crmdown command.

■ To stop the CRM from OS/390 MVS:

Run a CRMDOWN job using JCL written explicitly for your system. Refer to "Sample JCL for CRMDOWN Command" and Appendix A, "Command Reference Pages" for more information about sample CRMDOWN JCL and the crmdown command.

If the eAM gateway terminates abnormally, the persistent CRM feature allows the OS/390 MVS CRM to continue running. You must explicitly shutdown the CRM if you need to shut it down. The OS/390 MVS platform stops the CRM through the `crmdown` JCL.

# A Command Reference Pages

This section provides information about the commands an administrator uses to perform activities with the Communications Resource Manager (CRM) to maintain BEA eLink Adapter for Mainframe (eAM) applications. These commands work using JCL on OS/390MVS.

This section discusses the following commands:

■ CRMLOGS

■ crmlkon

■ crmlkoff

■ crmdown

■ SNACRM

## CRMLOGS

Displays the content and state of the Communications Resource Manager (CRM) log files in the current directory.

Synopsis     CRMLOGS <group> [<crm name>]

Description  Use the CRMLOGS command to display the contents and state of the two SNACRM log files. RSTRTLOG is the transaction state log used during the recovery process and the BLOBLOG log stores session and link information. Deleting the log files requires a cold start for each link involved.

CRMLOGS requires the following parameters:

group
>    SNA domain group name (required)

crm name
>    SNACRM name (default CRM)

Diagnostics    CRMLOGS exits with a return code of 0 upon successful completion.

Examples    To display the RSTRTLOG log file for group2, type:

CRMLOGS GROUP2 SNACRM.GROUP2.RSTRTLOG

To display the BLOBLOG log file for group1, type:

CRMLOGS GROUP1 SNACRM.GROUP1.BLOBLOG

See Also    SNACRM

# crmlkon

Start one or more named CRM links.

## Synopsis

crmlkon -n<hostname:port> [-v -i -h -u<keyfile>] <linkname> ...

## Description

crmlkon starts all of the CRM links named on the command line. This command is useful if one or more individual links failed to start when the CRM server booted. Use this command from any machine located on the same TCP/IP network as the machine running the CRM server. When used in a script, crmlkon will return 0 if the command could be sent to the target CRM. It will return 1 if the command could not be sent to the target CRM.

-n
>    Names the machine and port running the CRM server; indicates that the parameters immediately following are the hostname and port of the machine running the CRM where the linkname is located.

`-v`

> Specifies verbose. Normally the command will not produce any messages, facilitating use in a script.

`-i`

> Ignores errors. When specifying multiple links, any error encountered when issuing CRM commands will cause crmlkon to stop processing links and return. Errors can be ignored for individual links and processing continued with the next named link.

`-u`

> Establishes that process authentication is in effect for communications between this process and the CRM.

`<keyfile>`

> Is the location of a file containing a hash key known to both this process and the CRM. The file contains a single line specifying a unique hash key (limited to eight characters). The file should be protected.

`<linkname>`

> Names the link to be started. This is the SNALINKS entry in the eAM gateway (GWSNAX) configurations that defines this link. Multiple link names can be specified.

`<hostname:port>`

> Specifies the IP host name and port of the machine running the CRM where the `linkname` is located.

`h`

> Invokes a help screen that shows the syntax usage for this command.

## Example

To start links `link2` and `cicstest` owned by the CRM running on `mach1` at port 5000:

```
crmlkon -n mach1:5000 link2 cicstest
```

## Diagnostics

crmlkon only checks the syntax of the command. Separate facilities, either jsnacrm or mainframe-based facilities must be used to determine if the link actually became active. If the command could not be successfully sent to the CRM, crmlkon prints an error message if in verbose mode and exits with error code 1. Upon successful completion, crmlkon exits with exit code 0.

# crmlkoff

Stop one or more named CRM links.

## Synopsis

```
crmlkoff -n<hostname:port> [-v -i -h -u<keyfile>] <linkname> ...
```

## Description

crmlkoff stops all of the CRM links named on the command line. This is useful if one or more individual links need to be stopped after the CRM server booted. It can be used from any machine located on the same TCP/IP network as the machine running the CRM server. It can be used in a script and will return 0 if the command could be sent to the target CRM. It will return 1 if the command could not be sent to the target CRM.

-n

> Names the machine and port running the CRM server; indicates that the parameters immediately following are the hostname and port of the machine running the CRM where the linkname is located. This is required.

-v

> Specifies verbose. Normally the command will not produce any messages, facilitating use in a script.

-i

> Ignores errors. When specifying multiple links, any error encountered when issuing CRM commands will cause crmlkoff to stop processing links and return. Errors can be ignored for individual links, and processing continued with the next named link.

-u

Establishes that process authentication is in effect for communications between this process and the CRM.

<keyfile>

Is the location of a file containing a hash key known to both this process and the CRM. The file contains a single line specifying a unique hash key (limited to eight characters). The file should be protected.

<linkname>

Names the link to be stopped. This is the *DM_SNALINKS entry in the DMCONFIG which defines this link. Multiple link names can be specified.

h

Invokes a help screen that shows the syntax usage for this command.

## Example

To stop links `link1` and `cicstest` owned by the CRM running on `mach` at port 5000:

```
crmlkoff -n mach:5000 link1 cicstest
```

## Diagnostics

`crmlkoff` only checks the syntax of the command. Separate facilities, either `jsnacrm` or mainframe based facilities, must be used to determine if the link actually became inactive. If the command could not be successfully sent to the CRM, `crmlkoff` prints an error message if in verbose mode and exits with error code `1`. Upon successful completion, `crmlkoff` exits with exit code `0`.

# crmdown

Shut down a CRM.

## Synopsis

```
crmdown -n<hostname:port> [-v -i -h -u<keyfile>]
```

## Description

crmdown shuts down the CRM specified on the command line. It can be used from any machine located on the same TCP/IP network as the machine running the CRM server. It can be used in a script and will return 0 if the command could be sent to the target CRM. It will return 1 if the command could not be sent to the target CRM.

-n

    Names the machine and port running the CRM server.

-v

    Specifies verbose. Normally the command will not produce any messages, facilitating use in a script.

-i

    Ignores errors.

-u

    Establishes that process authentication is in effect for communications between this process and the CRM.

<keyfile>

    Is the location of a file containing a hash key known to both this process and the CRM. The file contains a single line specifying a unique hash key (limited to eight characters). The file should be protected.

## Example

To stop the CRM running on mach1 at port 5000:

```
crmdown -n mach1:5000
```

## Diagnostics

crmdown only checks the syntax of the command. If the command could not be successfully sent to the CRM, crmdown prints an error message if in verbose mode and exits with error code 1. Upon successful completion, crmdown exits with exit code 0.

# SNACRM

Launches the Communications Resource Manager.

| Synopsis | `SNACRM [ -t 0|1|2|3 ] [-p<nbr>] [-s] [-n <type>:#:#] [-u <keyfile>]`<br>`<addr> <group>` |
|---|---|

Description
: When you start SNACRM from the command line, the CRM Command Line Console puts its prompt in the window and, if exited, shuts down all of the active links.

    You must configure one CRM for each eAM gateway, as well as configure one stack for each CRM definition. Each stack can manage one or more SNA links.

    SNACRM has two types of log files stored in $APPDIR, RSTRTLOG, and BLOBLOG. RSTRTLOG is the transaction state log used during the recovery process, while the BLOBLOG log stores session and link information. Deleting the log files requires a cold start for each link involved. You can use the CRMLOGS command to display the contents and state of the SNACRM log files.

Trace Options
: When initiating the CRM from the UNIX command line, you can specify any of the following CRM trace levels using the -t option:

    ■ 0 = No tracing. Setting this level effectively disables SNACRM tracing and closes the trace file, if there is one. If tracing is subsequently restarted, a new file is created with an incremented numerical suffix.

    ■ 1 = Minimum tracing. At this level, SNACRM traces only major events and is sufficient only to determine the sequence of application conversations.

    ■ 2 = Medium tracing. At this level, SNACRM also traces all I/O buffers.

    ■ 3 = Maximum tracing. At this level, SNACRM also traces all APPC verbs.

    The APPC Stack API trace is either enabled or disabled using the -s option of the SNACRM command (refer to "General Options"). If enabled, it generally shows the parameters and results of all API calls. Depending on the stack being used, other options (such as vendor-specified environment variables) may have to be activated for SNACRM to enable the trace.

Authentication Option
: `-u`

    Establishes that process authentication is in effect for communications between the CRM and a client process.

    `keyfile`

    Is the location of a file containing a hash key known to both the CRM and the client process. The file contains a single line specifying a unique hash key (limited to eight characters). The file should be protected.

Encryption
Option

```
-n
```

Establishes that encryption is in effect for this process.

```
type
```

Is the encryption type. Currently, the only valid entry is GPE (for general purpose encryption).

```
min
```

Designates the minimum number of bits to be used for encryption. These levels are used during the negotiation between the CRM and client process. Any number is acceptable, but the negotiated values resolve to 0, 56, or 128.

```
max
```

Designates the maximum number of bits to be used for encryption. These levels are used during the negotiation between the CRM and client process. Any number is acceptable, but the negotiated values resolve to 0, 56, or 128

Performance
Option

The performance option is a tunable parameter used only for the OS/390 Unix and MVS CRM. These CRM versions use threads to process a request and this parameter indicates how many threads to start for processing SNA requests.

The performance option uses the following format:

```
-p <nbr>
```

where <nbr> equals the number of threads to start.

The default for this value is 100 threads. This should correspond to the load of SNA requests that will be made concurrently. If the number of requests exceeds the number of threads, the request will still be executed, however, the completion time could be affected.

It is recommended that you do not exceed 200 threads. The CRM is tuned for a maximum of 200 threads. It is recommended that you lower the 100 threads value if you have a restriction on the number of threads that can be active in your system.

General Options

The following parameters apply to this command:

```
-s
```

APPC Stack API trace (default none)

**Note:** On OS/390 platforms, the General Trace Facility (GTF) is used to capture API trace records under user EID 2EA. The GTF must be active on the OS/390 platform to use the `-s` parameter.

addr

Socket listening address (required)
//host:port *group*
GWSNAX Group Name (required)

Environment Variables

You must set the following environment variables before starting the CRM:

- From the UNIX command line:

```
FIEDLTBLS32 must contain fmb.def.
FLDTBLDIR32 must contain the path to the eAM libraries directory
(for example, eAM40/lib).
APPDIR must be set to the application directory.
```

- From MVS:

```
FLDTBLDIR32=DD:FLDTBL
FIELDTBLS32=FMB
APPDIR=<High level qualifier for datasets to be created in
APPDIR>
```

Portability

Refer to the *BEA eLink Adapter for Mainframe Release Notes* for a list of operating systems and stacks that can communicate with the CRM. (See the list of the requirements that are needed to SNACRM itself on the OS/390 platform.)

Interoperability

SNACRM is interactive with the following:

- CICS 4.1 through TS 1.3

- IMS 4.1 or higher

- MVS 5.22 9510 or higher

- OS/390 1.2 or higher

- VTAM for MVS/ESA, version 4.3 or higher

- CRM for OS/390 and MVS versions require SecureWay Communications Server, OS/390 2.8 or higher (see *BEA eLink Adapter for Mainframe Release Notes* for required IBM APAR for version 2.8)

Diagnostics

SNACRM exits with a return code of 0 upon successful completion.

Examples

Following is an example of the SNACRM command:

```
SNACRM -t 0 //myhost:5587 GROUP2 /dev/null>std.out 2>std.err &
```

When you start SNACRM from the UNIX command line, the following CRM Command Line Console appears:

```
$ SNACRM -t 0 //myhost:5587 GROUP2
      BEA eAM Resource Manager started Thu Dec 11
      18:40:49.098 1997
      [SNACRM]

      Console active. Enter commands
      ?>
       da => Display active tasks
       dl => Display remote links
       ds => Display link statistics
       dt => Display trace status
       st => Start all links
       sh => Stop all links and terminate
       si => Terminate immediately (no quiesce)
```

To launch SNACRM with the console running in the background:

```
$ SNACRM -t 0 //myhost:5587 GROUP2 <dev/null>std.out 2>std.err &
```

To launch SNACRM with detailed tracing and APPC Stack API tracing turned on from the command line using the host/port address, type:

```
SNACRM -t 2 -s //myhost:5587
```

# B Error Messages

The following Communications Resource Manager (CRM) messages can be displayed in the Tuxedo/eLink Platform/Weblogic Enterprise console window.

| 9001:ERROR | `<taskname>` timed out with failCode `<failcode>` | |
|---|---|---|
| | DESCRIPTION | A conversation has timed out in the CRM with the stack return code of `<failcode>`. A timer event set to watch a conversation has expired. |
| | | `<taskname>` may appear as: |
| | | OB-Conversation #*nn* (`<linkref>`) tx #*m* `<tranname>`, or |
| | | IB-Conversation #nn (`<linkref>`) tx #*m* `<tranname>` |
| | | where: |
| | | *nn* is an internal APPC conversation number. |
| | | *m* is the transaction context where -1 signifies non-transactional. |
| | ACTION | Examine `stderr` and the ULOG for additional information concerning the failure. |
| 9002:ERROR | Server (`<stackref>`) Creation Failed | |
| | DESCRIPTION | CRM was unable to instantiate the stack object due to an error. |
| | ACTION | Check for additional messages in `stderr`. The shared library for the stack or the stack interface might not have been loaded due to an incorrect library path. |

| 9003:ERROR | Server Failed (`<stackref>`), Code = `<returncode>` | |
|---|---|---|
| | DESCRIPTION | CRM received a bad return code from the stack start-up. |
| | ACTION | The `<returncode>` is the value returned by the SNA Stack software. Check the status of the stack, the configuration of the stack, and the gateway configuration. |
| | **Note:** | Unless you started the CRM as a Tuxedo server, you must manually kill the CRM process. |
| 9004:ERROR | Configuration change on link `<linkref>` requires cold start | |
| | DESCRIPTION | Attempting to do a warm start after changing the domain configuration. |
| | ACTION | Change start type to COLD and restart. |
| 9005:WARNING | Unrecovered transaction, ID=`<tctxt>`, blob dropped. Transaction presumed forgotten. | |
| | DESCRIPTION | An attempt was make by Tuxedo to recover the specified transaction that was unknown by the CRM. It is presumed that it was already committed or aborted prior to the recovery attempt. |
| | ACTION | None. This message is for information only. |
| 9006:ERROR | Unable to start the recovery task for link `<linkref>` | |
| | DESCRIPTION | An error occurred during the warm start of Tuxedo. |
| | ACTION | Cold start the Tuxedo application. |
| 9008:WARNING | Unknown tranid dropped, id=`<tctxt>` | |
| | DESCRIPTION | Recovery was requested by Tuxedo on a transaction that was already forgotten by the CRM. |
| | ACTION | None. This message is for information only. |

| 9009:ERROR | No blob with recovery request. Transaction dropped, id=`<tctxt>` | |
|---|---|---|
| | DESCRIPTION | Attempting to warm start after the CRM's `BLOBLOG` has been modified. |
| | ACTION | Change start type to COLD and restart. |
| 9010:ERROR | `<taskname>` failed with failCode `<failcode>` | |
| | DESCRIPTION | A conversation has failed with the stack return code of `<failcode>`. |
| | | `<taskname>` may appear as: |
| | | OB-Conversation #*nn* (`<linkref>`) tx #*m* `<tranname>`, or |
| | | IB-Conversation #nn (`<linkref>`) tx #*m* `<tranname>` |
| | | where: |
| | | *nn* is an internal APPC conversation number. |
| | | *m* is the transaction context where -1 signifies non-transactional. |
| | | Possible values for the `<failcode>` are: |
| | | 1. Communications - unable to create the APPCserver object. |
| | | 2. MemoryAllocation - internal error allocating memory. |
| | | 3. InvalidObject - a CRM object could not be created or has been made invalid by some previous error. |
| | | 4. InputOutput - error occurred during file I/O or an unexpected APPC return code was received. |
| | | 5. Registration - internal task cannot be registered. |
| | ACTION | Examine `stderr` and the `ULOG` for additional information concerning the failure. For failcode Input/Output, verify that the user starting the CRM process has the proper file permissions for the `BLOBLOG` and `RSTRTLOG`. If no apparent error is found, contact BEA Customer Support. |

| 9011:ERROR | Attempt to connect as second master refused! | |
|---|---|---|
| | DESCRIPTION | A second eAM gateway is attempting to connect to the CRM as a master gateway. Only one master gateway is allowed. |
| | ACTION | Ensure that multiple Tuxedo configurations do not use the same CRM address. |
| 9012:ERROR | Attempt to connect as master in autonomous mode refused! | |
| | DESCRIPTION | An attempt to connect to the CRM as a master gateway was made when the CRM was running in autonomous mode. |
| | ACTION | Ensure that multiple Tuxedo configurations do not use the same CRM address. |
| 9013:ERROR | Attempt to connect with incorrect group name (`<groupname>`) refused! | |
| | DESCRIPTION | The group name in the gateway configuration file does not match the group name specified in the CRM command line. |
| | ACTION | Correct the group name that is in error and restart. |
| 9014:ERROR | INTERNAL ERROR: memory allocation failed [for new context/data buffer] | |
| | DESCRIPTION | Internal error allocating memory. No more memory. |
| | ACTION | Contact BEA Customer Support. |
| 9015:ERROR | INTERNAL ERROR: server registration failed | |
| | DESCRIPTION | Internal error registering the APPC server. APPC libraries not found. The stack failed. |
| | ACTION | Contact BEA Customer Support. |

| 9016:ERROR | Link refers to undefined APPC stack (`<stackref>`)! | |
|---|---|---|
| | DESCRIPTION | The `stackref` in the link configuration is incorrect. |
| | ACTION | Correct the `stackref` that is in error, run `dmloadcf`, and restart. |
| 9017:ERROR | INTERNAL ERROR: link registration failed | |
| | DESCRIPTION | Internal error registering the link. The stack failed. |
| | ACTION | Contact BEA Customer Support. |
| 9018:ERROR | Invalid Transition <additional information> | |
| | DESCRIPTION | An internal state table failure has occurred. The `<additional information>` will be one of the following: 1. From `<oldstate>` to `<newstate>` for `<dir>`-bound transaction #`<tid>` 2. To `<newstate>` for inbound transaction #`<tid>` 3. To `<newstate>` for outbound transaction #`<tid>` |
| | ACTION | Contact BEA Customer Support. |
| 9019:ERROR | Unknown Service Correlator = `<correlator>`, message dropped | |
| | DESCRIPTION | Internal error assigning service correlator values. Message context lost. |
| | ACTION | Contact BEA Customer Support. |
| 9020:ERROR | Duplicate Service Correlator = `<correlator>` | |
| | DESCRIPTION | Internal error assigning service correlator values. |
| | ACTION | Contact BEA Customer Support. |

| 9021:ERROR | Invalid Remote Link Name `<linkref>` | |
|---|---|---|
| | DESCRIPTION | The remote link name in a request does not match any defined link name. |
| | ACTION | Correct the gateway configuration and restart. |
| 9022:ERROR | Invalid transaction context = `<tctxt>` | |
| | DESCRIPTION | Internal error assigning transaction context values. Bad transaction. |
| | ACTION | Contact BEA Customer Support. |
| 9023:ERROR | Unknown Service Correlator = `<correlator>`, message dropped | |
| | DESCRIPTION | Internal error assigning service correlator values. Message context lost. |
| | ACTION | Contact BEA Customer Support. |
| 9024:ERROR | Invalid initial syncpoint received from subordinate, tx#`<tid>` | |
| | DESCRIPTION | Syncpoint processing protocol violation. Subordinate member of conversation attempted to initiate a syncpoint. XA does not support syncpoints from subordinate members. |
| | ACTION | Contact BEA Customer Support. |
| 9025:ERROR | Invalid Input Message Discarded | |
| | DESCRIPTION | Internal error, bad message sent between eAM gateway and CRM. Possibly incompatible eAM gateway and CRM. |
| | ACTION | Contact BEA Customer Support. |
| 9026:ERROR | CNOS Notification Received for unknown partner `<partnerLU>` | |
| | DESCRIPTION | Multiple instances of the CRM may be using the same local LU. |
| | ACTION | Ensure that multiple Tuxedo configurations do not use the same local LU. |

| 9027:WARNING | Remote Stop Received for `<linkref>` | |
|---|---|---|
| | DESCRIPTION | The remote host has issued a stop for the specified link. |
| | ACTION | None. This message for information only. |
| 9028:WARNING | Remote Start Received for `<linkref>` | |
| | DESCRIPTION | The remote host has issued a start for the specified link. |
| | ACTION | None. This message for information only. |
| 9029:ERROR | Undefined Remote LU on link `<linkref>` | |
| | DESCRIPTION | The remote LU does not exist as defined. |
| | ACTION | Check the gateway configuration file and the stack configuration and correct the mis-match. |
| 9030:ERROR | Unable to start session on link `<linkref>`. Reason=`<reason>` | |
| | DESCRIPTION | Link activation failure due to SNA error. |
| | ACTION | `<reason>` is the description of the stack return code. Determine the cause and correct. |
| 9031:ERROR | Unable to initialize link `<linkref>`. Reason=`<reason>` | |
| | DESCRIPTION | Link initialization failure due to SNA error. |
| | ACTION | `<reason>` is the description of the stack return code. Determine the cause and correct. |
| 9032:ERROR | No Available Session on link `<linkref>` for context `<correlator>` | |
| | DESCRIPTION | Max sessions has been exceeded. |
| | ACTION | Check session limits in gateway configuration, stack configuration, CICS or VTAM. Increase if necessary. |

| 9033:ERROR | Requested Synclevel not supported by link `<linkref>` for context `<correlator>` (synclevel `<level>`) | |
| --- | --- | --- |
| | DESCRIPTION | Attempted to issue a request at sync level `<level>` on a link that does not support that level. |
| | ACTION | Correct application or gateway configuration. |
| 9034:ERROR | Service Request at SyncLevel=2 Rejected on PENDING link `<linkref>` for context `<correlator>` | |
| | DESCRIPTION | An attempt to start a new sync level 2 request has been received and the Link is currently processing recovery information. |
| | ACTION | Wait until recovery is complete to request sync level 2 services. |
| 9035:ERROR | Inbound Request Transform Failed (`<status>`) for context `<correlator>` | |
| | DESCRIPTION | An error has occurred while processing the CICS transform for an inbound DPL request. This normally occurs when the API entry in the gateway configuration for the local service specifies CICS instead of ATMI. |
| | ACTION | Check gateway configuration for incorrect specification of local service API entry. |
| 9036:ERROR | Inbound Response Transform Failed (`<status>`) for context `<correlator>` | |
| | DESCRIPTION | An error has occurred while processing the CICS transform for an inbound DPL response. This normally occurs when the API entry in the gateway configuration for the local service specifies CICS instead of ATMI. |
| | ACTION | Check gateway configuration for incorrect specification of local service API entry. |

| 9037:ERROR | Outbound Request Transform Failed (`<status>`) for context `<correlator>` | |
|---|---|---|
| | DESCRIPTION | An error has occurred while processing the CICS transform for an outbound DPL request. This normally occurs when the API entry in the gateway configuration for the remote service specifies CICS instead of ATMI. |
| | ACTION | Check gateway configuration for incorrect specification of local service API entry. |
| 9038:ERROR | Outbound Response Transform Failed (`<status>`) for context `<correlator>` | |
| | DESCRIPTION | An error has occurred while processing the CICS transform for an outbound DPL response. This normally occurs when the API entry in the gateway configuration for the remote service specifies CICS instead of ATMI. |
| | ACTION | Check gateway configuration for incorrect specification of local service API entry. |
| 9039:ERROR | Conversation terminated without confirm for context `<correlator>` | |
| | DESCRIPTION | Sync level 2 conversation was terminated with out confirm. |
| | ACTION | Check application program and correct. |
| 9040:ERROR | Inbound Confirm not supported | |
| | DESCRIPTION | Host application is requesting an inbound confirm. This is not supported. |
| | ACTION | Check host application program and correct. |
| 9041:ERROR | Inbound Confirm for multi-ISRT not supported | |
| | DESCRIPTION | Host IMS application is requesting an inbound confirm and using multiple ISRT commands. This is not supported. |
| | ACTION | Check host application program and correct. |

| 9043:ERROR | Missing send last from host (ATMI request/response) for context `<correlator>` | |
|---|---|---|
| | DESCRIPTION | Host application did not issue send last during an outbound request/response service. The host application may have abended. |
| | ACTION | Check application program and correct. |
| 9044:INFO | DPL program abended with CICS code `<abendcode>`, program=`<progname>` | |
| | DESCRIPTION | The specified host DPL program has abended with the code specified. |
| | ACTION | None. This message is for information only. |
| 9045:INFO | DPL program failed with CICS rcode `<eibrcode>`, program=`<progname>` | |
| | DESCRIPTION | The specified host DPL program has failed with the `eibrcode` specified. |
| | ACTION | None. This message is for information only. |
| 9046:ERROR | Invalid combination for Service Context `<correlator>`, `<combination>` | |
| | DESCRIPTION | The specified `<combination>` is invalid. It will be one of the following: 1. Sync-Level, function, and API 2. Function and API |
| | ACTION | Examine the gateway configuration and make corrections. |
| 9047:ERROR | Sequence number error for Service Context `<correlator>`, seqno `<seqno>` | |
| | DESCRIPTION | There has been a sequence number failure for the specified context. Context is out of sequence. |
| | ACTION | Contact BEA Customer Support. |

| 9048:ERROR | Invalid conversation task for Service Context `<correlator>`, task=`<task>` | |
|---|---|---|
| | DESCRIPTION | The conversation has already been terminated. |
| | ACTION | Contact BEA Customer Support. |
| 9049:ERROR | Invalid task switch for Service Context `<correlator>`, from `<task1>` to `<task2>` | |
| | DESCRIPTION | An internal protocol violation has occurred. |
| | ACTION | Contact BEA Customer Support. |
| 9050:ERROR | Transformer creation failed for inbound transaction `<trancode>` | |
| | DESCRIPTION | An internal error has occurred. Possibly out of memory. |
| | ACTION | Contact BEA Customer Support. |
| 9051:ERROR | Transformer failed for inbound transaction `<trancode>` | |
| | DESCRIPTION | An internal error has occurred. Resource name is not present. Mainframe compatibility problem. |
| | ACTION | Contact BEA Customer Support. |
| 9052:WARNING | Inter-task Message dropped (`<verbname>`), parm=`<parm>` From: `<task1>` to `<task2>` | |
| | DESCRIPTION | An internal message between two tasks has been dropped. |
| | ACTION | None. This message is for information only. |
| 9053:ERROR | Attempt to send `<nnnnn>` bytes (> 32767) | |
| | DESCRIPTION | The length of a send request exceeded 32767 (including overhead). |
| | ACTION | Check application program and correct. |

| 9054:ERROR | Allocation Failure for `<trancode>` on `<remotesysid>`: `<error>` | |
| --- | --- | --- |
| | DESCRIPTION | An Allocation error occurred. |
| | ACTION | The reason for the failure is described by <error>. Correct problem with configuration or application. |
| 9055:ERROR | Invalid Exchange Logs GDS variable received from `<remotesysid>` | |
| | DESCRIPTION | The log files for the CRM have been incorrectly modified. |
| | ACTION | Run CRMLOGS to examine the CRM log file. Cold start the Tuxedo application. |
| 9056:ERROR | Invalid cold start received from `<remotesysid>`. Unrecovered local transactions are pending. | |
| | DESCRIPTION | Attempting to cold start host while warm starting Tuxedo. |
| | ACTION | Run CRMLOGS to examine the CRM log file. Cold start the Tuxedo application. |
| 9057:ERROR | Invalid warm start received from `<remotesysid>`. Unknown log name. | |
| | DESCRIPTION | The log files for the CRM have been incorrectly modified. |
| | ACTION | Run CRMLOGS to examine the CRM log file. Cold start the Tuxedo application. |
| 9058:ERROR | Invalid Compare States GDS variable received from `<remotesysid>` | |
| | DESCRIPTION | The log files for the CRM have been incorrectly modified. |
| | ACTION | Run CRMLOGS to examine the CRM log file. Cold start the Tuxedo application. |

| 9059:ERROR | Mixed Heuristic on link `<linkref>` for `<unitofwork>` Correlator [`<correlator>`] | |
|---|---|---|
| | DESCRIPTION | One side has reported committed while the other side has reported aborted. |
| | ACTION | Check the `ULOG` for any additional messages. |
| 9060:WARNING | Inbound Exchange Logs Rejected for `<remotesysid>` | |
| | DESCRIPTION | Link not configured for sync level 2. |
| | ACTION | None. This message is for information only. |
| 9061:WARNING | Link `<linkref>` not configured for sync level 2 | |
| | DESCRIPTION | Link specified by `<linkref>` is not configured for sync level 2. |
| | ACTION | None. This message is for information only. |
| 9062:ERROR | Exchange Logs Rejected for `<remotesysid>`, Restart Type or Log Name Mismatch | |
| | DESCRIPTION | The log files for the CRM have been incorrectly modified. |
| | ACTION | Run `CRMLOGS` to examine the CRM log file. Cold start the Tuxedo application. |
| 9063:ERROR | Exchange Logs failed with `<linkref>` | |
| | DESCRIPTION | An error occurred during the exchange logs process. |
| | ACTION | Run `CRMLOGS` to examine the CRM log file. Cold start the Tuxedo application. |
| 9064:ERROR | Invalid initial syncpoint received from subordinate, `%s` | |
| | DESCRIPTION | An internal error has occurred during the commit process. XA does not support syncpoints originating from subordinate members. |
| | ACTION | Contact BEA Customer Support |

| 9069 ERROR | SNACRM encryption setup failed | |
|---|---|---|
| | DESCRIPTION | An error has occurred while establishing link-level encryption with the SNACRM. |
| | ACTION | Verify that the setup of encryption on both sides of the link is correct. Verify that there are common encryption levels in the ranges specified on the process command lines. Verify that the correct encryption libraries are installed |
| 9072 ERROR | Attempted access by unauthorized SNACRM client | |
| | DESCRIPTION | A client has attempted to access the SNACRM without the proper authentication or encryption setup. |
| | ACTION | Verify that the client should have access to the SNACRM. Verify that encryption is set up correctly in both the SNACRM and the client, and that the correct security add-on packages are installed. Verify that the authentication file is set up correctly, and that both the CRM and the client have correct access privileges. |
| 9073 ERROR | SNACRM authentication setup failed | |
| | DESCRIPTION | An error has occurred while authenticating a SNACRM client. |
| | ACTION | Verify that the authentication file(s) for both the SNACRM and the client are set up correctly, and that both have correct access privileges. Verify that the authentication file is specified correctly on both command lines. |

# Glossary

**A**

**Access Control Lists (ACL)**

A Tuxedo security feature that controls client access to services by means of lists that are automatically checked each time a service is requested.

**ACID Properties**

The essential characteristic of transaction processing systems:
*Atomicity*: All changes that a transaction makes to a database are made permanent, or else are nullified.
*Consistency*: A successful transaction transforms a database from a previous valid state to a new valid state.
*Isolation*: Changes that a transaction makes to a database are not visible to other operations until the transaction completes its work.
*Durability*: Changes that a transaction makes to a database survive future system or media failures.

**Application**

A BEA Tuxedo System/T *application* is bounded by the environment described in a single TUXCONFIG file. In /Domain, a BEA Tuxedo System/T application can communicate with another application via a domain gateway group.

**Application Domain**

When used alone, the term *Domain* can mean a number of things. In order to avoid confusion, the term *application domain* is used to refer to a BEA Tuxedo application bounded by the configuration of a tmconfig file. This application domain can be restricted to a single platform or shared memory (SHM) environment, or it can be scaled across multiple machines in a multiple processor (MP) environment.

**Application Programming Interface (API)**

1) The verbs and environment that exist at the application level to support a par-

ticular system software product. 2) A set of code that enables a developer to initiate and complete client/server requests within an application. 3) A set of calling conventions that define how to invoke a service. A set of well-defined programming interfaces (entry points, calling parameters, and return values) by which one software program utilizes the services of another

**Application Program-to-Program Communication (APPC)**

An interface to LU6.2 services; provides a set of primitives to conduct conversations in LU6.2 sessions.

**C**

**Client**

A program designed to request information from a server.

**CNOS**

*CNOS* are service programs implemented as part of an LU6.2. The CNOS programs negotiate session limits between the two communication LU.

**Common Programming Interface for Communications (CPI-C)**

An interface to LU6.2 services. It is a simpler set of primitives than the APPC interface and is intended for use in program-to-program communications.

**Communication Resource Manager (CRM)**

A process that provides all of the sync-level two logic for an SNA domain gateway and directly communicates with the PU2.1 server.

**Customer Information Control System/Enterprise System Architecture (CICS/ESA)**

An operating environment devised by IBM that provides a foundation upon which to write customer applications programs. Several facilities useful for programming are supplied by the CICS environment, including basic mapping services (BMS), transient data queues (TD), temporary storage files (TS), memory services, etc. Customer applications are built as separate transaction programs, and are invoked as transactional tasks. CICS/ESA is a trademark of International Business Machines (IBM), Inc.

**D**

**Distributed Program Link (DPL)**

Function of CICS ISC that supports LINK requests between CICS regions, and is similar to a BEA Tuxedo request/response.

**Distributed Transaction Processing (DTP)**

A CICS intercommunication in which processing is distributed among transactions that communicate synchronously over intersystem or inter-region links. It is roughly equivalent to BEA Tuxedo conversations.

**E**

**ESA**

(ESA) Enterprise Systems Architecture is the conceptual structure and functional behavior of the latest range of IBM mainframe computers. ESA/370 is the fourth step in an evolution of which the first three steps were System/360, System/370, and System/370 extended architecture (370-XA).

**F**

**Field Manipulation Language (FML)**

A set of C language functions for defining and manipulating storage structures called field buffers. Cooperating processes can send and receive data in fielded buffers.

**FML Buffer**

A buffer of self-describing data items accessed through the Field Manipulation Language (FML) API.

**I**

**Inbound**

A generic term referring to request message direction relative to the server or a response message direction relative to the client.

**Information Management System (IMS)**

A database manager used by CICS/ESA to allow access to data. IMS provides for the arrangement of data in an hierarchical structure and a common access approach in application programs that manipulate IMS databases.

**InterSystem Communications (ISC)**

Communication between separate systems by means of SNA networking facilities or by means of the application-to-application facilities. ISC links CICS systems to other systems and may be used for communication between user applications or to transparently execute CICS functions on a remote CICS system.

**L**

**Logical Unit (LU)**

In SNA, a port through which a user gains access to the services of a network. Also, see System Network Architecture (SNA).

**LU6.2**

*LU6.2* is a particular SNA logical unit that identifies a specific set of services for program to program communication. Services include syncpoint, mapping of buffers into records, message confirmation, and security.

**M**

**MODENAME**

MODENAME is a configuration parameter that names a set of characteristics for a group of BEA eLink Java Adapter for Mainframe sessions. In the CICS region, the mode is defined in VTAM and referenced in CICS and the DMCONFIG file.

**mirror task**

CICS/ESA task that services incoming requests that specify a *mirror transaction* (CSMI, CSM1, CSM2, CSM3, CSM5, CPMI, CVMI, or a user-defined mirror transaction identifier).

**mirror transaction**

CICS/ESA transaction that recreates a request that is function shipped from one system to another, issues the request on the second system, and passes the acquired data back to the first system.

**Mirror Transaction Identifier Support**

BEA eLink Java Adapter for Mainframe feature which enables BEA Tuxedo clients to invoke host CICS/ESA programs and, conversely, CICS/ESA client programs to invoke BEA Tuxedo services. Based on the IBM CICS/ESA mirror transaction.

**Multiple Virtual Storage (MVS)**

An operating system for processing systems consisting of one or more mainframe processors.

**O**

**Outbound**

A generic term referring to request message direction relative to the client or response message direction relative to the server.

**P**

**PU 2.1**

PU 2.1 is an SNA server that provides client programs with access to an SNA network. The PU 2.1 Server's functions include managing the physical connections, providing SNA node device emulation, servicing Logical Units, communicating with client programs, and controlling and monitoring the local SNA resources.

**R**

**Resource Definition Online (RDO)**

The recommended method of defining resources to CICS/ESA. Resource definitions are created interactively by a CEDA transaction, or by the DFHCSDUP utility. Both methods store definition in the CICS/ESA system definition data set (CSD). At CICS initialization, CSD definitions are selectively installed as CICS system tables controlled by a user-supplied list of definitions. CEDA-defined resource definitions can be installed while CICS is active and used immediately.

**S**

**Server**

A computer or program that is dedicated to providing information in response to external requests.

**Session**

When two LU bind with each other, that is, when they have successfully negotiated how they will communicate, they are said to be in *session*. SNA has fixed limits on the number of sessions configured for an LU type.

**SNA Communication Resource Manager (CRM)**

A process that provides all of the sync-level two logic for an SNA domain gate-

way and directly communicates with the PU2.1 server.

**Stack**

Platform vendor-supplied software that provides connectivity to an SNA network.

**Synchronization Level (sync level)**

The level of synchronization (0, 1, or 2) established for an APPC session between intercommunicating CICS/ESA transactions. Level 0 gives no synchronization support, level 1 allows the exchange of private synchronization requests, and level 2 gives full CICS/ESA synchronization support, with backout of all updates to recoverable resources if failure occurs.

**System Network Architecture (SNA)**

A seven-layer networking protocol. Each layer of the protocol has a set of associated data communication services. The services of the uppermost layer are embodied in a Logical Unit (LU). Each LU type defined in SNA has its own specific set of services available to an end user for communicating. The end user may be a terminal device, or an application program. The SNA structure enables the end user to operate independently, unaffected by the specific facilities used for information exchange.

**T**

**Transaction**

1) A complete unit of work that transforms a database from one consistent state to another. In DTP, a transaction can include multiple units of work performed on one or more systems.

2) A logical construct through which applications perform work on shared resources (e.g., databases). The work done on behalf of the transaction conforms to the four ACID properties: atomicity, consistency, isolation, and durability.

**Transaction Processing (TP)**

A form of immediate data processing in which user requests are entered directly to the terminal and on-line programs satisfy the requests; for example, by updating database files and displaying output messages.

**V**

**Virtual Telecommunications Access Method (VTAM)**

A set of programs that control communication across a network between terminals and application programs.

# Index