



# **BEA eLink™ Adapter for Mainframe**

## **CRM Administration Guide**

Version 4.1  
Document Date: December 2001

# Copyright

Copyright © 2001 BEA Systems, Inc. All Rights Reserved.

## Restricted Rights Legend

This software and documentation is subject to and made available only pursuant to the terms of the BEA Systems License Agreement and may be used or copied only in accordance with the terms of that agreement. It is against the law to copy the software except as specifically allowed in the agreement. This document may not, in whole or in part, be copied photocopied, reproduced, translated, or reduced to any electronic medium or machine readable form without prior consent, in writing, from BEA Systems, Inc.

Use, duplication or disclosure by the U.S. Government is subject to restrictions set forth in the BEA Systems License Agreement and in subparagraph (c)(1) of the Commercial Computer Software-Restricted Rights Clause at FAR 52.227-19; subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013, subparagraph (d) of the Commercial Computer Software--Licensing clause at NASA FAR supplement 16-52.227-86; or their equivalent.

Information in this document is subject to change without notice and does not represent a commitment on the part of BEA Systems. THE SOFTWARE AND DOCUMENTATION ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. FURTHER, BEA Systems DOES NOT WARRANT, GUARANTEE, OR MAKE ANY REPRESENTATIONS REGARDING THE USE, OR THE RESULTS OF THE USE, OF THE SOFTWARE OR WRITTEN MATERIAL IN TERMS OF CORRECTNESS, ACCURACY, RELIABILITY, OR OTHERWISE.

## Trademarks or Service Marks

BEA, Jolt, Tuxedo, and WebLogic are registered trademarks of BEA Systems, Inc. BEA Builder, BEA Campaign Manager for WebLogic, BEA eLink, BEA Manager, BEA WebLogic Commerce Server, BEA WebLogic E-Business Platform, BEA WebLogic Enterprise, BEA WebLogic Express, BEA WebLogic Integration, BEA WebLogic Personalization Server, BEA WebLogic Portal, BEA WebLogic Server and How Business Becomes E-Business are trademarks of BEA Systems, Inc.

All other trademarks are the property of their respective companies.

BEA® eLink™ Adapter for Mainframe CRM Administration Guide

Part Number	Date	Software Version
N/A	December 2001	4.1

---

# Contents

## About This Document

What You Need to Know .....	vii
e-docs Web Site .....	viii
Related Information.....	viii
Documentation Conventions .....	viii
Contact Us .....	x

## 1. Understanding the Communications Resource Manager

About the Communications Resource Manager .....	1-2
System Configuration.....	1-3
Local Configuration .....	1-3
Distributed Configurations .....	1-4
Preparing Mainframe Configurations for CRM Requirements .....	1-5
Requirements for Establishing a VTAM Configuration .....	1-6
Configuring the CRM for a Third-Party Stack.....	1-6
Third-Party Stack Configuration.....	1-7
VTAM Cross Platform Definitions .....	1-12
Configuring the CRM for z/OS or OS/390 Platforms .....	1-14
Configuring the CICS/ESA LU .....	1-15
Putting It All Together.....	1-16
Example of Windows or UNIX Cross-Platform Definitions .....	1-17
Example of z/OS or OS/390 Definitions.....	1-21

## 2. Administering the SNA Components

Starting the CRM.....	2-1
Starting the CRM on UNIX .....	2-2
CRM Command .....	2-2

---

Command Line Options .....	2-3
Examples .....	2-5
Diagnostics .....	2-6
Starting the CRM on z/OS or OS/390 MVS .....	2-6
Sample JCL for the CRM Command .....	2-6
Using the CRM Monitor .....	2-8
Launching the CRM Monitor from the Windows Desktop .....	2-9
Launching the CRM Monitor from the Command Line .....	2-9
Activating and De-Activating Links .....	2-12
crmlkon Command .....	2-13
Command Line Options .....	2-13
Example .....	2-14
Diagnostics .....	2-14
Sample JCL for the crmlkon Command .....	2-15
crmlkoff Command .....	2-17
Command Line Options .....	2-17
Example .....	2-18
Diagnostics .....	2-18
Sample JCL for the crmlkoff Command .....	2-18
Reviewing CRM Log Files .....	2-20
CRMLOGS Command .....	2-20
Command Line Options .....	2-20
Example .....	2-21
Diagnostics .....	2-21
Sample JCL for the CRMLOGS Command .....	2-21
Stopping the CRM .....	2-23
crmdown Command .....	2-23
Command Line Options .....	2-24
Example .....	2-24
Diagnostics .....	2-24
Sample JCL for the CRMDOWN Command .....	2-25

## **A. Command Reference Pages**

CRM .....	A-1
CRMLOGS .....	A-5

---

crmlkon.....	A-6
crmlkoff .....	A-7
crmdown .....	A-8

## **B. Sample JCL for an MVS Platform**

Sample JCL for the CRM Command .....	B-2
Sample JCL for the crmlkon Command .....	B-3
Sample JCL for the crmlkoff Command .....	B-5
Sample JCL for the CRMDOWN Command .....	B-7
Sample JCL for the CRMLOGS Command .....	B-9

## **C. CRM Error Messages**

### **Glossary**

### **Index**



---

# About This Document

This document explains how the BEA eLink Adapter for Mainframe 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 the Tuxedo 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.
- [“Administering the SNA Components,”](#) describes how to use the CRM for administering communications resources.
- [“Command Reference Pages,”](#) lists and describes CRM system commands.
- [“Sample JCL for an MVS Platform,”](#) provides samples of JCL for running the CRM commands on an MVS operating system.
- [“CRM 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 applications.

---

# 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 eLink Adapter for Mainframe:

- *BEA eLink Adapter for Mainframe Release Notes*
- *BEA eLink Adapter for Mainframe Installation Guide*
- *BEA eLink Adapter for Mainframe CRM Administration Guide*
- *BEA eLink Adapter for Mainframe User Guide*
- *BEA eLink Adapter for Mainframe Reference Guide*

## Documentation Conventions

The following documentation conventions are used throughout this document:

Convention	Item
<b>boldface text</b>	Indicates terms defined in the glossary.
<a href="#">blue text</a>	Indicates hypertext links in PDF documents.
Ctrl+Tab	Indicates that you must press two or more keys simultaneously.



Convention	Item
<i>italics</i>	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. <i>Examples:</i> #include <iostream.h> void main ( ) the pointer psz chmod u+w * \tux\data\ap .doc tux.doc BITMAP float
<b>monospace boldface text</b>	Identifies significant words in code. <i>Example:</i> void <b>commit</b> ( )
<i>monospace italic text</i>	Identifies variables in code. <i>Example:</i> String <i>expr</i>
UPPERCASE TEXT	Indicates device names, environment variables, and logical operators. <i>Examples:</i> LPT1 SIGNON 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. <i>Example:</i> buildobjclient [-v] [-o name ] [-f <i>file-list</i> ]... [-l <i>file-list</i> ]...
	Separates mutually exclusive choices in a syntax line. The symbol itself should never be typed.

---

Convention	Item
...	Indicates one of the following in a command line: <ul style="list-style-type: none"><li>■ That an argument can be repeated several times in a command line</li><li>■ That the statement omits additional optional arguments</li><li>■ That you can enter additional parameters, values, or other information</li></ul> The ellipsis itself should never be typed. <i>Example:</i> <code>buildobjclient [-v] [-o name ] [-f file-list]... [-l file-list]...</code>
. . .	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 eLink Adapter for Mainframe 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 BEA eLink Adapter for Mainframe Communications Resource Manager, Document Edition 4.1.

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”
- “Preparing Mainframe Configurations for CRM Requirements”
- “Putting It All Together”

**Note:** All references to Application-to-Transaction Monitor Interface (ATMI) files, functions, and documentation apply to Tuxedo files, functions, and documentation.

# About the Communications Resource Manager

The Communications Resource Manager (CRM) is the component of the eLink Adapter for Mainframe 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 uses both SNA and TCP communication protocols. TCP protocols always flow between the eLink Adapter for Mainframe Gateway and CRM. SNA protocols always flow from the CRM to the mainframe and from either an SNA stack or VTAM on the mainframe to the CRM, depending upon the configuration option.

If the CRM is not running on a mainframe, it must run on the same platform as the SNA stack, but it may run on a different platform from the ATMI system and the eLink Adapter for Mainframe Gateway. For a complete list of supported operating systems, refer to the *BEA eLink Adapter for Mainframe Release Notes*.

The eLink Adapter for Mainframe Gateway provides the configuration for the CRM. The CRM should always be started and monitoring the address specified in the eLink Adapter for Mainframe Gateway configuration before the Gateway is brought up.

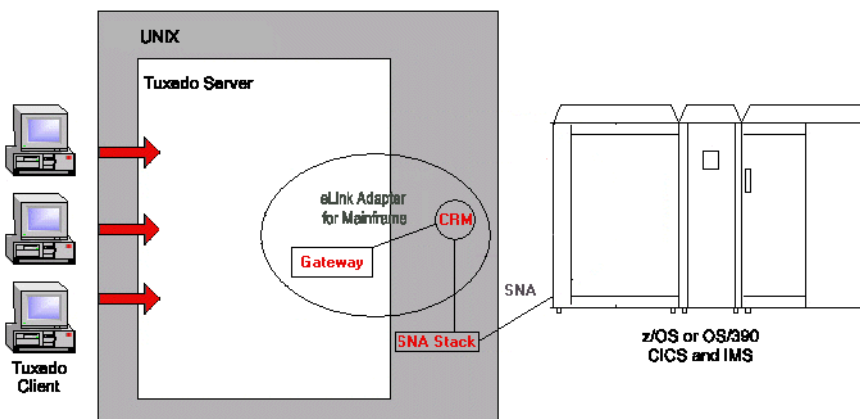
# System Configuration

The eLink Adapter for Mainframe system may be configured as either a local configuration or a distributed configuration running on a combination of two operating systems: UNIX, Windows, or a mainframe operating system. For a complete list of operating systems, refer to the *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. For this version of eLink Adapter for Mainframe, Solaris 8 is the only non-mainframe platform for which the CRM is available.

## Local Configuration

The local configuration combines the eLink Adapter for Mainframe Gateway and the CRM with the stack (PU2.1 server) on the same UNIX machine. It employs the IBM proprietary SNA protocol for transactions with the mainframe via the stack.

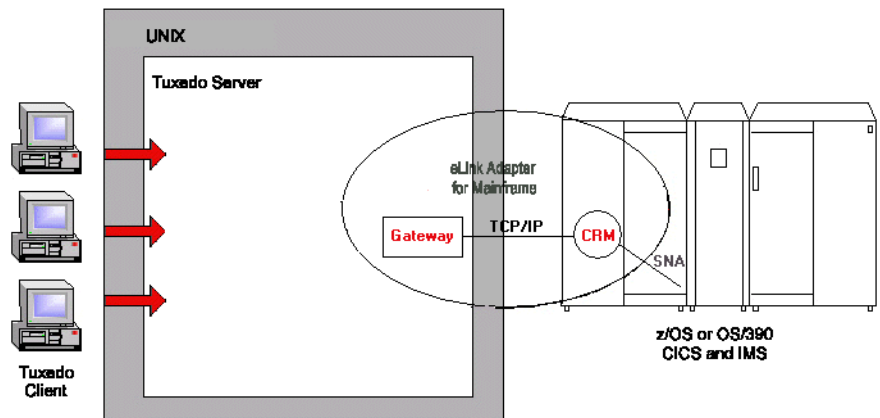
**Figure 1-1 eLink Adapter for Mainframe Local Configuration**



## Distributed Configurations

In one type of distributed configuration, the CRM is installed on the mainframe. If the CRM is installed on the mainframe, there is no need to install a third-party SNA stack. However, you must specify your Virtual Telecommunications Access Method (VTAM) in the eLink Adapter for Mainframe Gateway configuration file. [Figure 1-2](#) illustrates the distributed configuration architecture.

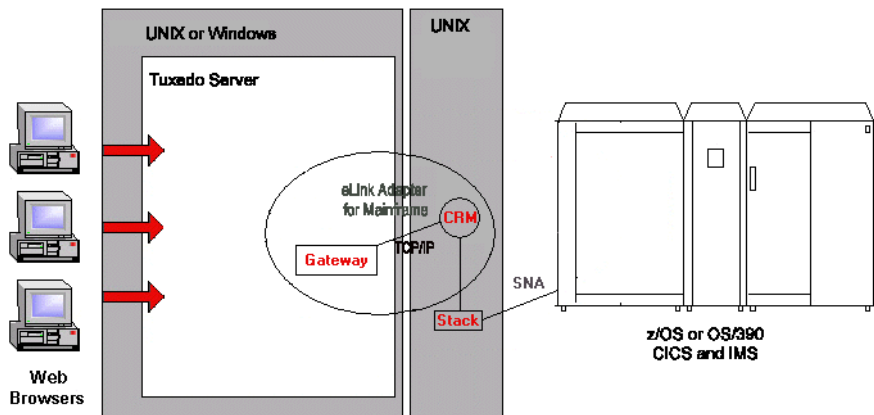
**Figure 1-2 eLink Adapter for Mainframe Distributed Configuration**





Another type of distributed configuration separates the CRM from the eLink Adapter for Mainframe Gateway on different UNIX or Windows machines. It employs Transmission Control Protocol/Internet Protocol (TCP/IP) connectivity between the eLink Adapter for Mainframe Gateway and the CRM, as well as SNA connectivity to the mainframe environment(s). This configuration gives you the flexibility to deploy the Tuxedo server separately from the CRM for installations that require the Tuxedo server 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 eLink Adapter for Mainframe Gateway and the remote CRM.

**Figure 1-3 Non-Mainframe eLink Adapter for Mainframe Distributed Configuration**



# Preparing Mainframe Configurations for CRM Requirements

The CRM runs as a separate native process that provides an emulation allowing CICS/ESA and IMS protocols to flow into and out of the ATMI environment. After you install the eLink Adapter for Mainframe product, you must set up your mainframe configurations to allow the CRM to access your CICS or IMS systems.

If the CRM runs on the mainframe, it uses the native SNA networking product, Virtual Telecommunications Access Method (VTAM), to establish its SNA connectivity to the CICS or IMS application environments. If the CRM runs on a UNIX platform, a third-party SNA stack must also be installed on this platform and configured to establish SNA connections with the CRM and target mainframe.

The topics in the following sections cover mainframe configuration requirements for the CRM.

## Requirements for Establishing a VTAM Configuration

The following requirements should be considered for establishing a VTAM configuration:

- If you are running your CRM on a mainframe, your mainframe must have a local VTAM configuration that communicates with the CRM.
- If you are using a third-party SNA stack, you must configure VTAM to communicate with the SNA stack.

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 eLink Adapter for Mainframe local domain.

## Configuring the CRM for a Third-Party Stack

A basic understanding of the mainframe configuration requirements provides a context for understanding the CRM functions and configuration requirements. The following sections discuss the various configuration considerations and give examples of those configurations.

**Note:** Consult with your local mainframe system administrator for specific information about your system. The examples in the following sections illustrate a starting point for configuring your system and do not represent all possibilities. The examples represent one way a mainframe can be configured to work in an Advanced Peer-to-Peer Networking (APPN) Local Area Network (LAN) environment on a SNAP-IX stack for Solaris 8.

## Third-Party Stack Configuration

Third-party stack configurations may be set up using stack-specific configuration utilities or by manually creating a configuration file with any text editor. [Listing 1-1](#) shows an example of an SNA node configuration file. The configuration file is divided into sections for various components of the configuration. Each section defines a component using parameters (or keywords) and values. Some of these keywords and values will affect how your eLink Adapter for Mainframe configuration is defined.

### **Listing 1-1 SNA Node Configuration File**

---

```
[define_node_config_file]
major_version = 5
minor_version = 1
update_release = 1
revision_level = 25

[define_node]
cp_alias = dalsun8
description = ""
fqcp_name = BEALAN.DALSUN8
node_type = LEN_NODE
mode_to_cos_map_supp = YES
mds_supported = YES
node_id = <bea08276>
max_locates = 1500
dir_cache_size = 255
max_dir_entries = 0
locate_timeout = 0
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
```

```
cos_table_version = VERSION_0_COS_TABLES
send_term_self = NO
disable_branch_awareness = NO
cplu_syncpt_support = NO
cplu_attributes = NONE
dlur_support = NO
pu_conc_support = YES
nn_rar = 128
max_ls_exception_events = 0
ptf_flags = NONE
```

```
[define_ethernet_dlc]
dlc_name = ETHER0
description = ""
neg_ls_supp = YES
initially_active = NO
adapter_number = 0
lan_type = 802_3_DIX
```

```
[define_ethernet_port]
port_name = ETSAP0
description = ""
dlc_name = ETHER0
port_type = PORT_SATF
port_number = 0
max_rcv_btu_size = 1033
tot_link_act_lim = 64
inb_link_act_lim = 0
out_link_act_lim = 0
ls_role = LS_NEG
implicit_dspu_services = NONE
implicit_dspu_template = ""
implicit_ls_limit = 0
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 = 0x04
implicit_cp_cp_sess_support = NO
implicit_limited_resource = NO
implicit_deact_timer = 30
implicit_hpr_support = NO
implicit_link_lvl_error = NO
implicit_uplink_to_en = NO
effect_cap = 3993600
connect_cost = 0
```

```
byte_cost = 0
security = SEC_NONSECURE
prop_delay = PROP_DELAY_LAN
user_def_parm_1 = 128
user_def_parm_2 = 128
user_def_parm_3 = 128
initially_active = YES
window_inc_threshold = 1
test_timeout = 10
test_timer_retry = 5
xid_timer = 10
xid_timer_retry = 5
ack_timeout = 5000
p_bit_timeout = 5000
t2_timeout = 100
rej_timeout = 10
busy_state_timeout = 30
idle_timeout = 30
max_retry = 3

[define_ethernet_ls]
ls_name = DV10QSN8
description = ""
port_name = ETSAP0
adj_cp_name = P390.DALVS10
adj_cp_type = END_NODE
mac_address = <0200bea07004>
lsap_address = 0x04
auto_act_supp = NO
tg_number = 0
limited_resource = NO
solicit_sscp_sessions = YES
pu_name = DV10QSN8
disable_remote_act = NO
default_nn_server = NO
dspu_services = NONE
dspu_name = <0000000000000000>
dlus_name = <00000000000000000000000000000000>
bkup_dlus_name = <00000000000000000000000000000000>
hpr_supported = NO
hpr_link_lvl_error = NO
link_deact_timer = 30
use_default_tg_chars = YES
ls_attributes = SNA
adj_node_id = <00000000>
local_node_id = <00000000>
cp_cp_sess_support = NO
effect_cap = 3993600
connect_cost = 0
```

```
byte_cost = 0
security = SEC_NONSECURE
prop_delay = PROP_DELAY_LAN
user_def_parm_1 = 128
user_def_parm_2 = 128
user_def_parm_3 = 128
target_pacing_count = 7
max_send_btu_size = 1033
ls_role = USE_PORT_DEFAULTS
max_ifrm_rcvd = 0
dlus_retry_timeout = 0
dlus_retry_limit = 0
branch_link_type = NONE
adj_brnn_cp_support = ALLOWED
dddlu_offline_supported = NO
initially_active = NO
restart_on_normal_deact = NO
react_timer = 30
react_timer_retry = 65535
test_timeout = 10
test_timer_retry = 5
xid_timer = 10
xid_timer_retry = 5
ack_timeout = 5000
p_bit_timeout = 5000
t2_timeout = 100
rej_timeout = 10
busy_state_timeout = 30
idle_timeout = 30
max_retry = 3
```

```
[define_partner_lu]
plu_alias = C10QA2
description = ""
fqplu_name = P390.C10QA2
plu_un_name = C10QA2
parallel_sess_supp = YES
max_mc_ll_send_size = 0
conv_security_ver = NO
```

```
[define_partner_lu]
plu_alias = C10DEV2
description = ""
fqplu_name = P390.C10DEV2
plu_un_name = C10DEV2
parallel_sess_supp = YES
max_mc_ll_send_size = 0
conv_security_ver = NO
```

```
[define_local_lu]
lu_alias = luqsun8a
list_name = ""
description = ""
lu_name = LUQSUN8A
lu_session_limit = 0
pu_name = <0000000000000000>
nau_address = 0
default_pool = NO
syncpt_support = YES
lu_attributes = NONE
sscp_id = 0
disable = NO
sys_name = ""
timeout = 60
back_level = NO
```

```
[define_local_lu]
lu_alias = luqsun8b
list_name = ""
description = ""
lu_name = LUQSUN8B
lu_session_limit = 0
pu_name = <0000000000000000>
nau_address = 0
default_pool = NO
syncpt_support = YES
lu_attributes = NONE
sscp_id = 0
disable = NO
sys_name = ""
timeout = 60
back_level = NO
```

```
[define_mode]
mode_name = SMSNA100
description = ""
max_neg_sess_lim = 32767
plu_mode_session_limit = 10
min_conwin_src = 5
min_conloser_src = 5
auto_act = 5
receive_pacing_win = 4
max_receive_pacing_win = 0
default_ru_size = YES
max_ru_size_upp = 1024
max_ru_size_low = 0
cos_name = #CONNECT
```

```
[define_directory_entry]
resource_name = P390.DALVS10
resource_type = ENCP_RESOURCE
description = (Auto defined - remote node)
parent_name = <00000000000000000000000000000000>
parent_type = ENCP_RESOURCE

[define_directory_entry]
resource_name = P390.C10QA2
resource_type = LU_RESOURCE
description = ""
parent_name = P390.DALVS10
parent_type = ENCP_RESOURCE

[define_directory_entry]
resource_name = P390.DALVS10
resource_type = LU_RESOURCE
description = (Auto defined - default LU)
parent_name = P390.DALVS10
parent_type = ENCP_RESOURCE

[define_defaults]
description = ""
mode_name = SMSNA100
implicit_plu_forbidden = NO
specific_security_codes = NO
limited_timeout = 20
```

---

## **VTAM Cross Platform Definitions**

The examples of cross platform definitions discussed in the following sections allow the VTAM network to communicate with your stack using the SNA protocol.

### **XCA Major Node Defines the LAN Adapter for SYS1**

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



### Listing 1-2 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

VTAM Switched Major Node (SWNET) definitions define and link together physical units (PU) and logical units (LU). Each workstation connected to a network must be represented on the network as a PU and each PU can have one or more LUs. The VTAM SWNET definition defines how the PUs and LUs communicate with a server.

The switched network definition example in this section ([Listing 1-3](#)) specifies the VTAM PU, 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 1-3 SWNET Major Node**

---

```
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
```

---

## **Configuring the CRM for z/OS or OS/390 Platforms**

VTAM must be configured to allow the CRM to communicate with CICS or IMS using the SNA protocol.

The APPLID definition shown in [Listing 1-4](#) shows an example of a VTAM stack configuration to run the CRM on z/OS or OS/390.

## **Listing 1-4 APPLID Definition (z/OS or OS/390)**

---

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

---

## 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 establish the configuration, 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, 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 eLink Adapter for Mainframe 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)
```

### 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 eLink Adapter for Mainframe 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

# 1 *Understanding the Communications Resource Manager*

---

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 number of remote sessions should equal the SESSNBR.

## Installing Resource Definitions

To install the resource definitions, put them on the host in a separate group. Use the CEDA INSTALL command.

For example:

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

## Viewing Connection and Session Status

After you have installed the resource 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
```

# Putting It All Together

The topics in the following sections cover examples of configurations for cross-platform definitions. These examples are for reference only and do not represent all configuration possibilities. Consult with your system administrator for specific information about your system.

## Example of Windows or UNIX Cross-Platform Definitions

Before installing eLink Adapter for Mainframe software, review [Table 1-1](#) for an example of Windows or UNIX cross-platform definitions. Consult with your VTAM system administrator to obtain the value indicated in the *Name* column and make the corresponding entries shown in the *Needed In* column.

**Note:** The DCL-based stack referred to in [Table 1-1](#) is the Solaris SNAP-IX stack.

[Listing 1-5](#) shows the corresponding eLink Adapter for Mainframe Gateway configuration file. Refer to the *BEA eLink Adapter for Mainframe User Guide* for information about setting up the Gateway configuration.

**Table 1-1 Summary of DCL Definitions**

Name	Originates In	Needed In
SNA Network ID (e.g. <b>SNANET1</b> ) and VTAM Host ID (e.g. <b>VTAMHOST</b> )	VTAM configuration	<b>DCL-based Stack Configuration:</b> Example: fqcp_name= <b>SNANET1</b> .SPARC1 adj_cp_name= <b>SNANET1</b> . <b>VTAMHOST</b> fqplu_name= <b>SNANET1</b> .CICSSYN
Mode Name (e.g. <b>SNA62</b> )	VTAM-MODEENT definition	<b>CICS Sessions Definition:</b> Example: MODENAME ( <b>SNA62</b> )  <b>DCL-based Stack Configuration:</b> Example: mode_name= <b>SNA62</b>  <b>GWSNAX Configuration:</b> Example: DM_SNALINKS MODENAME=" <b>SNA62</b> "
Control Point Name CPNAME (e.g. <b>SPARC1</b> )	VTAM-PU definition	<b>DCL-based Stack Configuration:</b> Example: fqcp_name=SNANET1.SPARC cp_alias= <b>SPARC1</b>

# 1 *Understanding the Communications Resource Manager*

**Table 1-1 Summary of DCL Definitions**

Name	Originates In	Needed In
Local LU Name (e.g. <b>L0F0024A</b> )	VTAM-LU definition	<b>CICS CONNECTION definition:</b> Example: NETNAME( <b>L0F0024A</b> ) <b>DCL-based Stack Configuration:</b> Example: lu_name= <b>L0F0024A</b> lu_alias= <b>L0F0024A</b> <b>GWSNAX Configuration:</b> Example: DM_SNASTACKS LOCALLU=" <b>L0F0024A</b> "
CICS LU Name (e.g. <b>CICSSYN</b> )	VTAM-LU definition	<b>DCL-based Stack Configuration:</b> Example: fqplu_name=SNANET1. <b>CICSSYN</b> plu_alias= <b>CICSSYN</b> <b>GWSNAX Configuration:</b> Example: DM_SNALINKS RLUNAME=" <b>CICSSYN</b> "
Terminal Identifier (e.g. <b>05DF0024</b> )	VTAM (IDNUM+IDBLK)	<b>DCL-based Stack Configuration:</b> Example: node_id=< <b>05000002</b> >
SYNCLVL	Stacks	<b>DCL-based Stack Configuration:</b> Example: [define local_lu] syncpt_support=CONFIRM <b>GWSNAX Configuration:</b> Example: DM_SNALINKS MAXSYNCLVL=1

**Table 1-1 Summary of DCL Definitions**

Name	Originates In	Needed In
Map all incoming conversations to eLink Adapter for Mainframe Gateway (make sure TPs have all privileges available, e.g. CNOS, service conversations, etc.).	Stacks	<b>DCL-based Stack Configuration:</b> Example: Sna_tps <404040...hex representation of 64 EBCDIC spaces...404040> TYPE=QUEUED TIMEOUT=-1 USERID=authorized_user_here GROUP=authorized_group_here LUALIAS=LOF0024A  <b>GWSNAX Configuration:</b> Example: DM_LOCAL_SERVICES RNAME="DPL1SVR"
CICS DPL program name (e.g. TOUPPER)	CICS/ESA	<b>GWSNAX Configuration:</b> Example: DM_REMOTE_SERVICES RNAME=TOUPPER

**Listing 1-5 Corresponding DMCONFIG File**

```
#=====

*DM_LOCAL_DOMAINS
simpsnad      GWGRP=GROUP2
              TYPE=SNAX
              DOMAINID="simpsnad"
              BLOB_SHM_SIZE=1000000
              DMTLOGDEV=/home/qctest/dmlog

*DM_REMOTE_DOMAINS

SIMPSNAG      TYPE=SNAX
              DOMAINID="SIMPSNAG"

*DM_SNACRM

simpcrm       SNACRMADDR="//myhost:6000
              NWDEVICE="/dev/tcp"
              LDOM="simpsnad"
```

# 1 *Understanding the Communications Resource Manager*

---

```
*DM_SNASTACKS

simpstk          SNACRM="simpcrm"
                  STACKTYPE="SPX70"
                  LOCALLU="L0F0024A"
                  LTPNAME="*"
                  STACKPARMS="myhost"

*DM_SNALINKS

simplk1          STACKREF="simpstk"
                  RDOM="SIMPSNAG"
                  LSYSID="BEA"
                  RSYSID="TEST"
                  RLUNAME="CICSSYN"
                  MODENAME="SNA62"
                  SECURITY="LOCAL"
                  STARTTYPE="COLD"
                  MAXSESS=10
                  MINWIN=5
                  MAXSYNCLVL=2

*DM_LOCAL_SERVICES

DPL1SVR          LDOM="simpsnad"
                  CONV=N
                  RNAME="DPL1SVR"
                  INBUFTYPE="STRING"
                  OUTBUFTYPE="STRING"

*DM_REMOTE_SERVICES

TOUPPER          AUTOTRAN=N
                  LDOM="simpsnad"
                  RDOM=SIMPSNAG
                  CONV=N
                  RNAME="TOUPPER"
                  INBUFTYPE="STRING"
                  OUTBUFTYPE="STRING"
                  FUNCTION="DPL"

*DM_ROUTING
```

---



## Example of z/OS or OS/390 Definitions

Before installing eLink Adapter for Mainframe software, review [Table 1-2](#) for an example of SNA definitions when the CRM runs on z/OS or OS/390 platforms. Consult with your system administrator to obtain the value indicated in the *Name* column and make the corresponding entries shown in the *Needed In* column.

[Listing 1-6](#) shows the corresponding eLink Adapter for Mainframe configuration file. Refer to the *BEA eLink Adapter for Mainframe User Guide* for information about setting up the Gateway configuration.

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

Name	Originates In	Needed In
Local LU Name (e.g. <b>BEAAPPL1</b> )	VTAM-LU definition	<b>CICS CONNECTION definition:</b> Example: NETNAME ( <b>BEAAPPL1</b> )  <b>VTAM Configuration:</b> Example: BEASNA VBUILD TYPE=APPL BEAAPPL1 APPL ACB=BEAAPPL1, APPC=YES, PARSESS=YES  <b>GWSNAX Configuration:</b> Example: DM_SNASTACKS LOCALLU= " <b>BEAAPPL1</b> "
Mode Name (e.g. <b>SNA62</b> )	VTAM-MODEENT definition	<b>CICS Sessions Definition:</b> Example: MODENAME ( <b>SNA62</b> )  <b>GWSNAX Configuration:</b> Example: DM_SNALINKS MODENAME= " <b>SNA62</b> "  <b>VTAM Configuration (not required):</b> Example: MODEENT= <b>SNA62</b>

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

Name	Originates In	Needed In
CICS LU Name (e.g. <b>CICSSYN</b> )	VTAM-LU definition	<b>GWSNAX Configuration:</b> Example: DM_SNALINKS RLUNAME= " <b>CICSSYN</b> "
SYNCLVL	VTAM-LU definition	<b>VTAM Configuration:</b> Example: SYNCLVL=CONFIRM  <b>GWSNAX Configuration:</b> Example: DM_SNALINKS MAXSYNCLVL=1
CICS DPL program name (e.g. TOUPPER)	CICS/ESA	<b>GWSNAX Configuration:</b> Example: DM_REMOTE_SERVICES RNAME=TOUPPER

**Listing 1-6 Corresponding DMCONFIG File**

---

```
#=====

*DM_LOCAL_DOMAINS
simpsnad      GWGRP=GROUP2
               TYPE=SNAX
               DOMAINID="simpsnad"
               BLOB_SHM_SIZE=1000000
               DMTLOGDEV=/home/qctest/dmlog

*DM_REMOTE_DOMAINS

SIMPSNAG      TYPE=SNAX
               DOMAINID="SIMPSNAG"

*DM_SNACRM

simpcrm       SNACRMADDR="//mymainframe:6000
               NWDEVICE="/dev/tcp"
               LDOM="simpsnad"

*DM_SNASTACKS
```

```
simpstk      SNACRM="simpcrm"  
             STACKTYPE="VTM28"  
             LOCALLU="BEAAPPL1"  
             LTPNAME="* "  
             STACKPARMS="mymainframe"
```

\*DM\_SNALINKS

```
simplk1      STACKREF="simpstk"  
             RDOM="SIMPSNAG"  
             LSYSID="BEA"  
             RSYSID="TEST"  
             RLUNAME="CICSSYN"  
             MODENAME="SMSNA100"  
             SECURITY="LOCAL"  
             STARTTYPE="COLD"  
             MAXSESS=10  
             MINWIN=5  
             MAXSYNCLVL=2
```

\*DM\_LOCAL\_SERVICES

```
DPL1SVR      LDOM="simpsnad"  
             CONV=N  
             RNAME="DPL1SVR"  
             INBUFTYPE="STRING"  
             OUTBUFTYPE="STRING"
```

\*DM\_REMOTE\_SERVICES

```
TOUPPER      AUTOTRAN=N  
             LDOM="simpsnad"  
             RDOM=SIMPSNAG  
             CONV=N  
             RNAME="TOUPPER"  
             INBUFTYPE="STRING"  
             OUTBUFTYPE="STRING"  
             FUNCTION="DPL"
```

\*DM\_ROUTING

---



# 2 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 applications.

The interface to the stack administration and configuration is dependent on the stack provider and is not 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”](#)
- [“Using the CRM Monitor”](#)
- [“Activating and De-Activating Links”](#)
- [“Reviewing CRM Log Files”](#)
- [“Stopping the CRM”](#)

## Starting the CRM

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 eLink Adapter for Mainframe Gateway.

You can start the CRM by:

- Entering the `CRM` command on the command line for UNIX.
- Running a CRM job using JCL written explicitly for your z/OS or OS/390 MVS system.
- Using the Tuexedo startup command, `tmboot` if you have a local configuration.

## Starting the CRM on UNIX

To ensure proper startup of the CRM, complete the following tasks. Use this method for starting the CRM only when you have a distributed configuration with the CRM on Solaris 8 or the mainframe.

1. Set `APPDIR` 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.

Enter the `CRM` command on the command line.

## CRM Command

The `CRM` command launches the Communications Resource Manager. 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.

You must configure one CRM for each eLink Adapter for Mainframe Gateway, as well as configure one stack for each CRM definition. Each stack can manage one or more SNA links.

CRM 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 CRM log files.

Example of the CRM command line:

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

Command Line Options

The following table provides descriptions of the valid options for the CRM command:

Keyword	Default	Required/ Optional	Description
-t [0 1 2 3]	0	Optional	<p>Turns tracing on and indicates the level of tracing.</p> <p><b>0=No tracing.</b></p> <p>Setting this level effectively disables CRM tracing and closes the trace file, if there is one. If tracing is subsequently restarted, a new file is created with an incremental numerical suffix. 0 is the default trace level.</p> <p><b>1=Minimum tracing.</b></p> <p>At this level, the CRM traces only major events and is sufficient only to determine the sequence of application conversations.</p> <p><b>2=Medium tracing.</b></p> <p>At this level, the CRM also traces all I/O buffers.</p> <p><b>3=Maximum tracing.</b></p> <p>At this level, the CRM also traces all APPC verbs.</p> <p><b>Note:</b> Trace options for the CRM may also be set from the CRM Monitor.</p>

## 2 Administering the SNA Components

---

Keyword	Default	Required/ Optional	Description
-s	Off (if parameter is not used)	Optional	<p>Indicates APPC Stack API trace is turned on</p> <p>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.</p> <p>If the APPC Protocol Stack API trace is 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 the CRM to enable the trace.</p> <p><b>Note:</b> Trace options for the APPC Stack API trace may also be set from the CRM Monitor.</p>
-p <nbr>	100 threads	Optional	<p>Turns on the performance option and indicates the number of threads to start.</p> <p>This value 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 is still executed; however, the completion time could be affected. Do not exceed 200 threads. The CRM is tuned for a maximum of 200 threads. Lower the threads value if you have a restriction on the number of threads that can be active in your system.</p>
<addr>	None	Required	<p>Specifies a TCP/IP address using //hostname:port_addr or the sockaddr_in format of family, port, address:</p> <p>&lt;0xFFFFPPPPAAAAAAA&gt;</p> <p>In this entry, arguments and options are defined in the following way:</p> <p>FFFF is the hex value of the protocol family, always 0x0002 for the INET family.</p> <p>PPPP is the hex value of an unused TCP/IP port.</p> <p>AAAAAAA is the hex value of the IP address for the machine running the CRM.</p>



Keyword	Default	Required/ Optional	Description
<group>	None	Required	Indicates the eLink Adapter for Mainframe Gateway Group Name.

## Examples

The following sections show three examples of different options for starting the CRM.

### Starting the CRM and CRM Command Line Console

To start a CRM from a UNIX command line, use a command similar to the following one:

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

When you start CRM from the UNIX command line, the following CRM command line console appears:

```
$ CRM -t 0 //myhost:5587 GROUP2
  A eLink Adapter for Mainframe Resource Manager started Thu Dec 11
    18:40:49.098 1997
    [CRM]

  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)
```

### Starting the CRM with CRM Command Line Console Running in Background

To launch the CRM with the console running in the background, use a command similar to the following one:

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

### Starting the CRM with Detailed Tracing and APPC Stack API Tracing

To launch the CRM with detailed tracing and APPC Stack API tracing turned on from the command line using the host/port address, use a command similar to the following one:

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

### Diagnostics

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

## Starting the CRM on z/OS or OS/390 MVS

The z/OS or 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.

```
APPDIR=<High level qualifier for datasets to be created in  
APPDIR>
```

2. Run a CRM job using JCL written for your system.

**Note:** If the CRM is installed on a z/OS or OS/390 MVS platform, it does not have to be restarted if the eLink Adapter for Mainframe Gateway goes down abnormally. A `tmshutdown` will not cause the CRM to shut down. You must run the `crmdown` utility to shut down the CRM. Only the z/OS or OS/390 MVS version and the z/OS or OS/390 UNIX version of the CRM have this persistent feature.

### Sample JCL for the CRM Command

This section provides an explanation of the SET commands and an example of JCL that can be used when you run the CRM command.

**SET STARTCMD**

Sets the CRM command line parameters.

**SET OBJLIB**

Indicates the name of the PDSE library where the CRM executable is installed.

**SET DATA**

Indicates the dataset containing the ENVFILE.

**SET ENVFILE**

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

**SET SIZE**

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

**SET ENV**

Indicates the ENVFILE DD name. This value is pre-set and should not be changed.

**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 CRM.

**Note:** Uncomment the SET CEE line and tailor the STEPLIB concatenation if these libraries are not in your system link library concatenation.

**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 2-1 Sample crmstart.jcl for CRM Command**

```
//*****
//* THIS JOB IS USED TO RUN THE CRM PROCESS. *
//* * *
//* @(#) $Id: crmstart.jcl,v 1.3 2001/05/07 23:41:27 crount Exp $ *
//* Copyright (c)2000 BEA Systems, Inc., all rights reserved. *
//*****
//* YOU MUST SET THE ENVIRONMENT VARIABLES NEEDED BY SNACRM *
//*****
//*****
//* USE THE SET STATEMENTS TO SET THE APPROPRIATE VALUES *
```

```
/* STARTCMD IS THE CRM COMMAND LINE *
/* OBJLIB IS THE LOAD LIBRARY CONTAINING THE PROGRAM EXECUTABLES *
/* DATA IS THE DATASET THAT CONTAINS THE ENVIRONMENT VARIABLES *
/* ENVFILE NAMES THE MEMBER THAT CONTAINS THE ENVIRONMENT VARS *
/* RUNOPTS SETS ANY DESIRED LE RUNTIME OPTIONS (OPTIONAL) *
/* SIZE SETS THE REGION SIZE FOR THE SNACRM PROCESS. 0M SETS NO *
/* LIMITS ON THE REGION SIZE *
/* TAILOR YOUR JCL FOR THE BELOW IF THESE LIBRARIES ARE NOT *
/* IN YOUR SYSTEM LINK LOAD LIBRARY CONCATENATION *
/* CEE IS THE HLQ FOR THE LE RUNTIME LIBRARY *
/* CBC IS THE HLQ FOR THE C/C++ RUNTIME LIBRARY *
/******
// SET STARTCMD=' " //<address>:<port>" <group>'
// SET OBJLIB=
// SET DATA=
// SET ENVFILE=ENV
// SET RUNOPTS=
// SET SIZE=0M
// SET ENV='ENVAR (" _CEE_ENVFILE=DD:ENV" ) '
/* SET CEE=CEE, CBC=CBC
//CRM EXEC PGM=CRM, REGION=&SIZE,
// PARM=' POSIX(ON) &ENV &RUNOPTS/&STARTCMD'
//STEPLIB DD DSN=&OBJLIB, DISP=SHR
/* DD DSN=&CEE..SCEERUN, DISP=SHR
/* DD DSN=&CBC..SCLBDLL, DISP=SHR
//MSGFILE DD SYSOUT=*
//TRACE DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//ENV DD DSN=&DATA (&ENVFILE) , DISP=SHR
//
```

---

## Using the CRM Monitor

The CRM Monitor is a Java application that allows you to connect to and monitor the CRM server through a graphical user interface (GUI). You can use the CRM Monitor to monitor link status and activity and start or stop diagnostic tracing of the CRM server. The CRM Monitor makes a network connection to the remote CRM server through a TCP/IP network connection. The CRM Monitor is installed with the Gateway, not on the mainframe.

## Launching the CRM Monitor from the Windows Desktop

If the CRM Monitor is installed on Windows, a short cut is created in the Start menu of the Desktop when you install eLink Adapter for Mainframe.

To launch the CRM Monitor from the Desktop:

- Choose Programs|BEA Weblogic E-Business Platform|eLink Adapter for Mainframe|CRM Monitor from the Start menu.

## Launching the CRM Monitor from the Command Line

The CRM Monitor can also be started from the command line on UNIX platforms.

To launch the CRM Monitor from the command line:

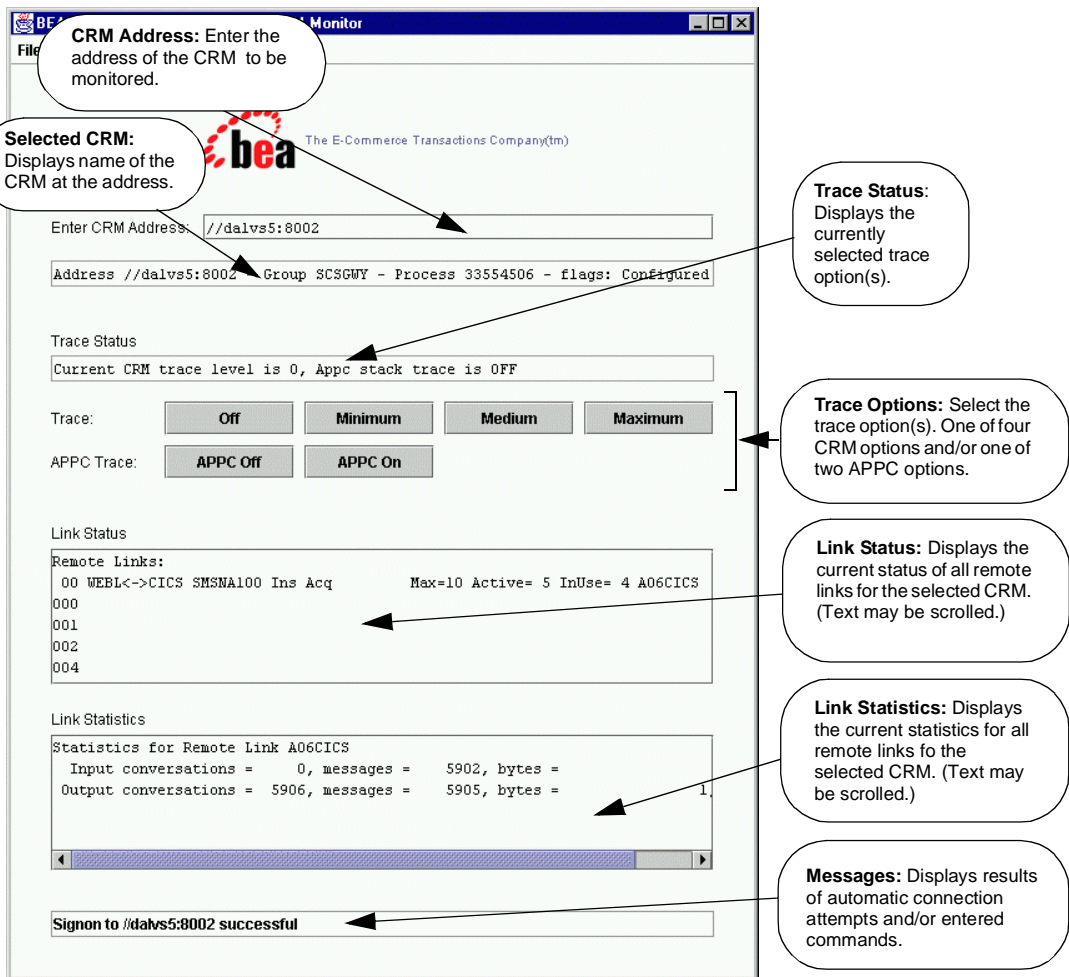
- Enter the following command for the Java 1.2.2 interpreter:  

```
java -jar crmmon.jar
```

## Setting CRM Monitor Options

Figure 2-1 shows an example of the CRM Monitor display.

**Figure 2-1 CRM Monitor Display**



After you launch the CRM Monitor, use the following instructions to set CRM Monitor options:

1. To monitor a CRM, type the CRM address in the **Enter CRM Address** text box using the following format:

`//host:port`

where `host` explicitly specifies the CRM host machine and `port` specifies the port number of the CRM on the CRM host machine.

This value may only be entered once per started instance of a CRM Monitor.

2. Select one of the following trace options for the CRM:

- **Stop CRM Trace** disables CRM tracing and closes the trace file, if it exists. This option is trace level 0 as described in the Trace Options section for the “CRM” command.
- **Set Minimum CRM Trace** establishes tracing of only major events. This level is sufficient only to determine the sequence of application conversations. This option is trace level 1 as described in the Trace Options section for the “CRM” command.
- **Set Medium CRM Trace** establishes tracing of major events plus tracing of all I/O buffers. This option is trace level 2 as described in the Trace Options section for the “CRM” command.
- **Set Maximum CRM Trace** establishes tracing of major events and all I/O buffers, plus tracing of all APPC verbs. This option is trace level 3 as described in the Trace Options section for the “CRM” command.

**Note:** The CRM Monitor does not show trace data. This data is captured in a file under the `APPDIR` directory of the CRM server (`APPDIR` is the variable name associated with the CRM directory). Please contact BEA Customer Support for help in locating the trace file(s) and interpreting them.

The time tag information in the CRM trace should reflect the current system time. In order to make use of the correct time zone information on UNIX and MVS systems, it is important that the `TZ` environment variable be set correctly. If this variable is not set correctly on your system, refer to your system documentation for further information.

3. Select one of the following APPC stack trace options:

- **Start APPC Stack Trace** establishes tracing of the APPC stack. This option 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. This option may be selected along

with any of the previous Trace Options. This option corresponds to the `-s` option as described in the “CRM” section.

- **Stop APPC Stack Trace** disables APPC stack trace, if established.

**Note:** Trace options for the CRM and APPC Stack API trace options may also be set from the CRM command. Refer to the “CRM” section for more information about trace options.

4. After you enter a CRM address and select trace options, the following fields display information about the CRM you are monitoring:

Display Field	Description
Selected CRM	Displays the name of the CRM at the address entered in the address field.
Trace Status	Displays the currently selected trace options.
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.)
Message Line	Displays messages showing either the results of automatic connection attempts or commands issued to change the trace options.

## Activating and De-Activating Links

You can activate and de-activate CRM links that have been defined in the `DM_SNALINKS` section of the `DMCONFIG` file by executing one of the link commands from the command line. There are two commands used to activate and de-activate links:

- “`crmlkon` Command”
- “`crmlkoff` Command”



You may also use Job Control Language (JCL) on a z/OS or OS/390 Multiple Virtual Storage (MVS) platform to set the environment and invoke link commands. The following sections provide descriptions of the link commands and samples of JCL that may be used for your MVS operating system.

# crmlkon Command

The `crmlkon` command starts one or more named CRM links.

`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. 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 returns 0 if the command could be sent to the target CRM. It returns 1 if the command could not be sent to the target CRM.

Example of the `crmlkon` command line follows:

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

Descriptions of the command line options follow.

## Command Line Options

The following table provides descriptions of the valid options for the `crmlkon` command:

Keyword	Default	Required/ Optional	Description
-n<hostname:port>	None	Required	Names the machine and port running the CRM server.
-v	Off	Optional	Specifies verbose. Normally the command will not produce any messages, facilitating use in a script.

## 2 Administering the SNA Components

---

Keyword	Default	Required/ Optional	Description
-i	Off	Optional	Ignores errors. When specifying multiple links, any error encountered when issuing CRM commands causes <code>crmlkon</code> to stop processing links and return. Errors can be ignored for individual links and processing continues with the next named link
<linkname>	None	Required	Names the link to be started. This is the <code>DM_SNALINKS</code> entry in the <code>gwsnax.cfg</code> that defines this link. Multiple link names can be specified.
-u<keyfile>	None	Optional	Specify a key file containing the authentication name to be used with the CRM.

### Example

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

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

### Diagnostics

`crmlkon` only checks the syntax of the command. Use the CRM Monitor to determine if the link actually became active. Refer to [“Setting CRM Monitor Options”](#) for more information. 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.

## Sample JCL for the crmlkon Command

This section provides an explanation of the SET commands and an example of JCL that can be used when you run the `crmlkon` command. 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.

SET LINKCMD	Sets the <code>crmlkon</code> command line parameters. Refer to “ <a href="#">crmlkon Command</a> ” for more information about the command line parameters.
SET OBJLIB	Indicates the name of the PDSE library where the <code>crmlkon</code> executable is installed.
SET DATA	Indicates the dataset containing the <code>ENVFILE</code> .
SET ENVFILE	Indicates the name of the PDS member that contains the environment variables for the <code>CRMLKON</code> . A sample member, <code>ENV</code> , is delivered with your product.
SET SIZE	Defines the region size for the running <code>crmlkon</code> task.
SET ENV	Indicates the <code>ENVFILE</code> DD name. This value is pre-set.
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 <code>crmlkon</code> .
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.

**Note:** Uncomment the `SET CBC` line and tailor the `STEPLIB` concatenation if these libraries are not in your system link library concatenation.

### Listing 2-2 Sample JCL for crmlkon Command

---

```
//*****
//*  THIS JOB IS USED FOR THE STAND-ALONE LINK COMMAND          *
//*  TO ACTIVATE A REMOTE LINK. SEE USER GUIDE FOR MORE INFO   *
//*                                                             *
//*  @(#) $Id: crmlkon.jcl,v 1.10 2001/05/07 23:41:27 crount Exp $ *
//*  Copyright (c)2000 BEA Systems, Inc., all rights reserved.  *
//*****
//*  YOU MUST SET THE ENVIRONMENT VARIABLES NEEDED BY CRMLKON  *
//*****
```

## 2 Administering the SNA Components

---

```
//*****
//* LINKCMD INDICATES THE DISTRIBUTED SNACRM ADDRESS AND LINKNAME*
//* OBJLIB IS THE LOAD LIBRARY CONTAINING THE ELINK ADAPTER FOR *
//* MAINFRAME PROGRAM OBJECTS *
//* RUNOPTS SETS ANY DESIRED LE RUNTIME OPTIONS *
//* DATA IS THE DATASET THAT CONTAINS THE ENVIRONMENT VARIABLES *
//* ENVFILE NAMES THE MEMBER THAT CONTAINS THE ENVIRONMENT VARS *
//* SIZE SETS THE REGION SIZE FOR THE SNACRM PROCESS *
//* TAILOR YOUR JCL FOR THE BELOW IF THESE LIBRARIES ARE NOT *
//* IN YOUR SYSTEM LINK LOAD LIBRARY CONCATENATION *
//* 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 DATA=
// SET ENVFILE=ENV
// 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
//ENV DD DSN=&DATA(&ENVFILE),DISP=SHR
//MSGFILE DD SYSOUT=*
//SYSPRINT DD SYSOUT=*

//
```

---

# crmlkoff Command

The `crmlkoff` command stops one or more named CRM links.

`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 returns 0 if the command could be sent to the target CRM. It returns 1 if the command could not be sent to the target CRM.

Example of the `crmlkoff` command line follows:

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

## Command Line Options

The following table provides descriptions of the valid options for the `crmlkoff` command:

Keyword	Default	Required/ Optional	Description
-n<hostname: port>	None	Required	Names the machine and port running the CRM server.
-v	Off	Optional	Specifies verbose. Normally the command will not produce any messages, facilitating use in a script.
-i	Off	Optional	Ignores errors. When specifying multiple links, any error encountered when issuing CRM commands causes <code>crmlkon</code> to stop processing links and return. Errors can be ignored for individual links and processing continues with the next named link
-h	None	Optional	Displays help for the command.
-u<keyfile>	None	Optional	Specifies a key file containing the authentication name to be used with the CRM.

Keyword	Default	Required/ Optional	Description
<linkname>	None	Required	Names the link to be stopped. This is the DM_SNALINKS entry in the DMCONFIG that defines this link. Multiple link names can be specified.

### Example

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

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

### Diagnostics

`crmlkoff` only checks the syntax of the command. Use the CRM Monitor to determine if the link actually became active. Refer to [“Setting CRM Monitor Options”](#) for more information. 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.

### Sample JCL for the crmlkoff Command

This section provides an explanation of the SET commands and an example of JCL that can be used when you run the `crmlkoff` command. 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.

SET LINKCMD	Sets the <code>crmlkoff</code> command line parameters. Refer to <a href="#">“Sample JCL for the crmlkon Command”</a> for more information about the command line parameters.
SET OBJLIB	Indicates the name of the PDSE library where the <code>crmlkoff</code> executable is installed.
SET DATA	Indicates the dataset containing the <code>ENVFILE</code> .
SET ENVFILE	Indicates the name of the PDS member that contains the environment variables for the <code>crmlkoff</code> . A sample member, <code>ENV</code> , is delivered with your product.

SET SIZE	Defines the region size for the running <code>crmlkoff</code> task.
SET ENV	Indicates the <code>ENVFILE</code> DD name.
SET CEE	Specifies the high-level qualifier for the Language Environment (LE) runtime library. CEE should be set to the prefix of the IBM LE data sets. Language Environment is required to run <code>crmlkoff</code> .
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.

**Note:** Uncomment the SET CBC line and tailor the STEPLIB concatenation if these libraries are not in your system link library concatenation.

### Listing 2-3 Sample JCL for crmlkoff Command

---

```
//*****
//* THIS JOB IS USED FOR THE STAND-ALONE LINK COMMAND          *
//* TO DEACTIVATE A REMOTE LINK. SEE USER GUIDE FOR MORE INFO *
//*                                                           *
//* @(#) $Id: crmlkoff.jcl,v 1.10 2001/05/07 23:41:27 crount Exp $*
//* Copyright (c)2000 BEA Systems, Inc., all rights reserved.  *
//*****
//* YOU MUST SET THE ENVIRONMENT VARIABLES NEEDED BY CRMLKOFF *
//*****
//*****
//* LINKCMD INDICATES THE DISTRIBUTED SNACRM ADDRESS AND LINKNAME*
//* OBJLIB IS THE LOAD LIBRARY CONTAINING THE ELINK ADAPTER FOR *
//* MAINFRAME PROGRAM OBJECTS                                  *
//* RUNOPTS SETS ANY DESIRED LE RUNTIME OPTIONS                *
//* DATA IS THE DATASET THAT CONTAINS THE ENVIRONMENT VARIABLES *
//* ENVFILE NAMES THE MEMBER THAT CONTAINS THE ENVIRONMENT VARS *
//* SIZE SETS THE REGION SIZE FOR THE SNACRM PROCESS           *
//* TAILOR YOUR JCL FOR THE BELOW IF THESE LIBRARIES ARE NOT   *
//* IN YOUR SYSTEM LINK LOAD LIBRARY CONCATENATION             *
//* 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 DATA=
// SET ENVFILE=ENV
// 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  
//ENV DD DSN=&DATA(&ENVFILE),DISP=SHR  
//MSGFILE DD SYSOUT=*  
//SYSPRINT DD SYSOUT=*  
//
```

---

# Reviewing CRM Log Files

You can display the content and state of the CRM log files by using the `CRMLOGS` command or `CRMLOGS JCL`.

## CRMLOGS Command

Use the `CRMLOGS` command to display the contents and state of the two CRM 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.

### Command Line Options

The following table provides descriptions of the valid options for the `CRMLOGS` command:

Keyword	Default	Required/ Optional	Description
group	None	Required	SNA domain group name.
crm name	Default CRM	Optional	Name of CRM.



## Example

To display the contents and state of the CRM log file for `dalvs5:8002` and `GROUP2`, use the following command:

```
CRMLOGS GROUP2 dalvs5:8002
```

## Diagnostics

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

## 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.

<code>SET LNKCMD</code>	Sets the <code>CRMLOGS</code> command line parameters.
<code>SET OBJLIB</code>	Indicates the name of the PDSE library where the <code>CRMLOGS</code> executable is installed.
<code>SET DATA1</code>	Indicates the name of the PDS library where the <code>CRMLOGS</code> required parameter file FMB was installed.
<code>SET DATA2</code>	Indicates the dataset containing the <code>ENVFILE</code> .
<code>SET ENVFILE</code>	Indicates the name of the PDS member that contains the environment variables for the <code>CRMLOGS</code> . A sample member, <code>ENV</code> , is delivered with your product.
<code>SET SIZE</code>	Defines the region size for the running <code>CRMLOGS</code> task.
<code>SET ENV</code>	Indicates the <code>ENVFILE</code> DD name.
<code>SET CEE</code>	Specifies the high-level qualifier for the LE runtime library. <code>CEE</code> should be set to the prefix of the IBM Language Environment data sets. Language Environment is required to run <code>CRMLOGS</code> .
<code>SET CBC</code>	Specifies the high-level qualifier for the C/C++ runtime library. <code>CBC</code> should be set to the prefix of the IBM C++ data sets.

### Listing 2-4 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 1.6 2001/05/07 23:41:27 crount Exp $ *
//* Copyright (c)2000 BEA Systems, Inc., all rights reserved. *
//*****
//* YOU MUST SET THE ENVIRONMENT VARIABLES NEEDED BY CRMLOGS *
//*****
//*****
//* SNACMD IS USED TO SET THE DESIRED SNACRM GROUP NAME      *
//* OBJLIB IS THE LOAD LIBRARY CONTAINING THE EAM PROGRAM OBJECTS *
//* RUNOPTS SETS ANY DESIRED LE RUNTIME OPTIONS (OPTIONAL)    *
//* DATA IS THE DATASET THAT CONTAINS THE ENVIRONMENT VARIABLES *
//* ENVFILE NAMES THE MEMBER THAT CONTAINS THE ENVIRONMENT VARS *
//* SIZE SETS THE REGION SIZE FOR THE SNACRM PROCESS         *
//* ENV SETS THE ENVIRONMENT VARIABLES DD NAME               *
//* TAILOR YOUR JCL FOR THE BELOW IF THESE LIBRARIES ARE NOT *
//* IN YOUR SYSTEM LINK LOAD LIBRARY CONCATENATION           *
//* CEE IS THE HLQ FOR THE LE RUNTIME LIBRARY                 *
//* CBC IS THE HLQ FOR THE C/C++ RUNTIME LIBRARY              *
//*****
// SET LOGSCMD=<group>
// SET OBJLIB=
// SET DATA=
// SET ENVFILE=ENV
// 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/&LOGSCMD'
//STEPLIB DD DSN=&OBJLIB,DISP=SHR
//* DD DSN=&CEE..SCEERUN,DISP=SHR
//* DD DSN=&CBC..SCLBDLL,DISP=SHR
//MSGFILE DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//ENV DD DSN=&DATA(&ENVFILE),DISP=SHR
//
```

---

# Stopping the CRM

The z/OS or OS/390 CRM will persist through termination of the Gateway, regardless of whether the Gateway was terminated by abnormal conditions or a `tmshutdown` command.

You can stop the CRM by:

- Entering the `crmdown` command on the command line for UNIX.
- Running a `CRMDOWN` job using JCL written explicitly for your z/OS or OS/390 MVS system.

## **crmdown Command**

The `crmdown` command shuts down the CRM specified on the command line.

`crmdown` 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 it returns 0 if the command could be sent to the target CRM. It returns 1 if the command could not be sent to the target CRM.

Example of the `crmdown` command line follows:

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

### Command Line Options

The following table provides descriptions of the valid options for the `crmdown` command:

Keyword	Default	Required/ Optional	Description
-n<hostname:port>	None	Required	Names the machine and port running the CRM server.
-v	Off	Optional	Specifies verbose. Normally the command will not produce any messages, facilitating use in a script.
-i	Off	Optional	Ignores errors. When specifying multiple links, any error encountered when issuing CRM commands causes <code>crmlkon</code> to stop processing links and return. Errors can be ignored for individual links and processing continues with the next named link
-h	None	Optional	Displays help for the command.
-u<keyfile>	None	Optional	Specifies a key file containing the authentication name to be used with the CRM.

### Example

To stop the CRM running on `mach1` at port 5000, use the following command:

```
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.

## Sample JCL for the CRMDOWN Command

This section provides an explanation of the `SET` commands and an example of JCL that can be used when you run the `crmdown` command.

**SET STOPCMD**

Sets the `crmdown` command line parameters.

**SET OBJLIB**

Indicates the name of the PDSE library where the `crmdown` executable is installed.

**SET DATA**

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.

**Note:** Uncomment the `SET CBC` line and tailor the `STEPLIB` concatenation if these libraries are not in your system link library concatenation.

### Listing 2-5 Sample JCL for crmdown Command

---

```
//*****
//* THIS JOB IS USED FOR THE STAND-ALONE COMMAND USED *
//* TO SHUTDOWN THE SNACRM PROCESS. SEE USER GUIDE FOR MORE INFO *
//*
//* @(#) $Id: crmdown.jcl,v 1.5 2001/05/07 23:41:27 crount Exp $ *
//* Copyright (c)2000 BEA Systems, Inc., all rights reserved. *
//*****
//* YOU MUST SET THE ENVIRONMENT VARIABLES NEEDED BY CRMDOWN *
//*****
//*****
//* STOPCMD INDICATES THE COMMAND LINE FOR CRMDOWN *
//* OBJLIB IS THE LOAD LIBRARY CONTAINING THE PROGRAM EXECUTABLES *
//* RUNOPTS SETS ANY DESIRED LE RUNTIME OPTIONS (OPTIONAL) *
//* DATA IS THE DATASET THAT CONTAINS THE ENVIRONMENT VARIABLES *
//* ENVFILE NAMES THE MEMBER THAT CONTAINS THE ENVIRONMENT VARS *
//* SIZE SETS THE REGION SIZE FOR THE SNACRM PROCESS *
//
//
// TAILOR YOUR JCL FOR THE BELOW IF THESE LIBRARIES ARE NOT
// IN YOUR SYSTEM LINK LOAD LIBRARY CONCATENATION
// CEE IS THE HLQ FOR THE LE RUNTIME LIBRARY
// CBC IS THE HLQ FOR THE C/C++ RUNTIME LIBRARY
//*****
// SET STOPCMD='-n<host name>:<port>'
// SET OBJLIB=
// SET RUNOPTS=
// SET DATA=
// SET ENVFILE=ENV
// 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/&STOPCMD'
//STEPLIB DD DSN=&OBJLIB,DISP=SHR
// * DD DSN=&CEE..SCEERUN,DISP=SHR
// * DD DSN=&CBC..SCLBDLL,DISP=SHR
//ENV DD DSN=&DATA(&ENVFILE),DISP=SHR
//MSGFILE DD SYSOUT=*
//SYSPRINT DD SYSOUT=*

//
```

---

# 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 applications. These commands work using JCL on z/OS or OS/390 MVS.

This section discusses the following commands:

- `CRM`
- `CRMLOGS`
- `crmlkon`
- `crmlkoff`
- `crmdown`

## CRM

Launches the Communications Resource Manager.

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

**Description**     When you start `CRM` 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 eLink Adapter for Mainframe Gateway, as well as configure one stack for each CRM definition. Each stack can manage one or more SNA links.

CRM 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 CRM 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 CRM 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, CRM traces only major events and is sufficient only to determine the sequence of application conversations.
- 2 = Medium tracing. At this level, CRM also traces all I/O buffers.
- 3 = Maximum tracing. At this level, CRM also traces all APPC verbs.

The APPC Stack API trace is either enabled or disabled using the `-s` option of the CRM 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 CRM to enable the trace.

**Note:** The time tag information in the CRM trace should reflect the current system time. In order to make use of the correct time zone information on UNIX and MVS systems, it is important that the `TZ` environment variable be set correctly. If this variable is not set correctly on your system, refer to your system documentation for further information.

**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:

`APPDIR` must be set to the application directory.

- From MVS:

`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 CRM itself on the OS/390 platform.)

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

**Examples**    Following is an example of the CRM command:

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

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

```
$ CRM -t 0 //myhost:5587 GROUP2
BEA eLink Adapter for Mainframe Resource Manager started Thu Dec 11
18:40:49.098 1997
[CRM]

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 CRM with the console running in the background:

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

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

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

# CRMLOGS

Displays the content and state of the BEA eLink Adapter for Mainframe 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 CRM 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**  
CRM 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 CRM.GROUP2.RSTRTLOG
```

To display the BLOBLOG log file for group1, type:

```
CRMLOGS GROUP1 CRM.GROUP1.BLOBLOG
```

See Also    CRM

# 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 DM\_SNALINKS entry in the DMCONFIG 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.





# B Sample JCL for an MVS Platform

Job Control Language (JCL) is used on a z/OS or 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 `CRM`, `CRMLKON`, `CRMLKOFF`, `CRMDOWN`, and `CRMLOGS` commands.

This section discusses the following topics:

- [“Sample JCL for the CRM Command”](#)
- [“Sample JCL for the crmlkon Command”](#)
- [“Sample JCL for the crmlkoff Command”](#)
- [“Sample JCL for the CRMDOWN Command”](#)
- [“Sample JCL for the CRMLOGS Command”](#)

# Sample JCL for the CRM Command

This section provides an explanation of the SET commands and an example of JCL that can be used when you run the CRM command.

SET STARTCMD	Sets the CRM command line parameters. Refer to “CRM” for more information about the command line parameters.
SET OBJLIB	Indicates the name of the PDSE library where the CRM executable is installed.
SET DATA	Indicates the dataset containing the ENVFILE.
SET ENVFILE	Indicates the name of the PDS member that contains the environment variables for the CRM. A sample member, ENV, is delivered with your product.
SET SIZE	Defines the region size for the running CRM task. The recommended setting for this option is OM to allow the CRM to start up and level out to the size it requires.
SET ENV	Indicates the ENVFILE DD name. This value is pre-set and should not be changed.
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 CRM.
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.

**Note:** Uncomment the SET CBC line and tailor the STEPLIB concatenation if these libraries are not in your system link library concatenation.

### Listing B-1 Sample JCL for CRM Command

---

```
//*****  
//* THIS JOB IS USED TO RUN THE CRM PROCESS. *  
//* * *  
//* @(#) $Id: crmstart.jcl,v 1.3 2001/05/07 23:41:27 crount Exp $ *  
//* Copyright (c)2000 BEA Systems, Inc., all rights reserved. *  
//*****  
//* YOU MUST SET THE ENVIRONMENT VARIABLES NEEDED BY CRM *  
//*****  
//*****
```

```
//* USE THE SET STATEMENTS TO SET THE APPROPRIATE VALUES      *
//* STARTCMD IS THE CRM COMMAND LINE                            *
//* OBJLIB IS THE LOAD LIBRARY CONTAINING THE PROGRAM EXECUTABLES*
//* DATA IS THE DATASET THAT CONTAINS THE ENVIRONMENT VARIABLES *
//* ENVFILE NAMES THE MEMBER THAT CONTAINS THE ENVIRONMENT VARS  *
//* RUNOPTS SETS ANY DESIRED LE RUNTIME OPTIONS (OPTIONAL)      *
//* SIZE SETS THE REGION SIZE FOR THE SNACRM PROCESS. 0M SETS NO *
//* LIMITS ON THE REGION SIZE                                   *
//* TAILOR YOUR JCL FOR THE BELOW IF THESE LIBRARIES ARE NOT    *
//* IN YOUR SYSTEM LINK LOAD LIBRARY CONCATENATION              *
//* CEE IS THE HLQ FOR THE LE RUNTIME LIBRARY                   *
//* CBC IS THE HLQ FOR THE C/C++ RUNTIME LIBRARY                *
//*****
// SET STARTCMD=' '//<address>:<port>" <group>'
// SET OBJLIB=
// SET DATA=
// SET ENVFILE=ENV
// SET RUNOPTS=
// SET SIZE=0M
// SET ENV=' ENVAR (" _CEE_ENVFILE=DD:ENV" ) '
//* SET CEE=CEE,CBC=CBC
//CRM      EXEC  PGM=CRM,REGION=&SIZE,
// PARM=' POSIX(ON) &ENV &RUNOPTS/&STARTCMD'
//STEPLIB DD DSN=&OBJLIB,DISP=SHR
//*        DD DSN=&CEE..SCEERUN,DISP=SHR
//*        DD DSN=&CBC..SCLBDLL,DISP=SHR
//MSGFILE DD SYSOUT=*
//TRACE DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//ENV DD DSN=&DATA(&ENVFILE),DISP=SHR
```

---

## Sample JCL for the crmlkon Command

This section provides an explanation of the SET commands and an example of JCL that can be used when you run the `crmlkon` command. 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.

**SET LINKCMD** Sets the `crmlkon` command line parameters. Refer to “[crmlkon](#)” for more information about the command line parameters.

## B Sample JCL for an MVS Platform

---

SET OBJLIB	Indicates the name of the PDSE library where the <code>crmlkon</code> executable is installed.
SET DATA	Indicates the dataset containing the <code>ENVFILE</code> .
SET ENVFILE	Indicates the name of the PDS member that contains the environment variables for the <code>CRMLKON</code> . A sample member, <code>ENV</code> , is delivered with your product.
SET SIZE	Defines the region size for the running <code>crmlkon</code> task.
SET ENV	Indicates the <code>ENVFILE</code> DD name. This value is pre-set.
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 <code>crmlkon</code> .
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.

**Note:** Uncomment the `SET CBC` line and tailor the `STEPLIB` concatenation if these libraries are not in your system link library concatenation.

### Listing B-2 Sample JCL for `crmlkon` Command

---

```
//*****
//* THIS JOB IS USED FOR THE STAND-ALONE LINK COMMAND *
//* TO ACTIVATE A REMOTE LINK. SEE USER GUIDE FOR MORE INFO *
//* *
//* @(#) $Id: crmlkon.jcl,v 1.10 2001/05/07 23:41:27 crount Exp $ *
//* Copyright (c)2000 BEA Systems, Inc., all rights reserved. *
//*****
//* YOU MUST SET THE ENVIRONMENT VARIABLES NEEDED BY CRMLKON *
//*****
//*****
//* LINKCMD INDICATES THE DISTRIBUTED SNACRM ADDRESS AND LINKNAME*
//* OBJLIB IS THE LOAD LIBRARY CONTAINING THE ELINK ADAPTER FOR *
//* MAINFRAME PROGRAM OBJECTS *
//* RUNOPTS SETS ANY DESIRED LE RUNTIME OPTIONS *
//* DATA IS THE DATASET THAT CONTAINS THE ENVIRONMENT VARIABLES *
//* ENVFILE NAMES THE MEMBER THAT CONTAINS THE ENVIRONMENT VARS *
//* SIZE SETS THE REGION SIZE FOR THE SNACRM PROCESS *
//* TAILOR YOUR JCL FOR THE BELOW IF THESE LIBRARIES ARE NOT *
//* IN YOUR SYSTEM LINK LOAD LIBRARY CONCATENATION *
//* 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 DATA=  
// SET ENVFILE=ENV  
// 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  
//ENV DD DSN=&DATA(&ENVFILE),DISP=SHR  
//MSGFILE DD SYSOUT=*  
//SYSPRINT DD SYSOUT=*  
//
```

---

## Sample JCL for the crmlkoff Command

This section provides an explanation of the SET commands and an example of JCL that can be used when you run the `crmlkoff` command. 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.

SET LINKCMD	Sets the <code>crmlkoff</code> command line parameters. Refer to “ <a href="#">crmlkoff</a> ” for more information about the command line parameters.
SET OBJLIB	Indicates the name of the PDSE library where the <code>crmlkoff</code> executable is installed.
SET DATA	Indicates the dataset containing the ENVFILE.
SET ENVFILE	Indicates the name of the PDS member that contains the environment variables for the <code>crmlkoff</code> . A sample member, <code>ENV</code> , is delivered with your product.
SET SIZE	Defines the region size for the running <code>crmlkoff</code> task.
SET ENV	Indicates the ENVFILE DD name.

## B Sample JCL for an MVS Platform

---

**SET CEE** Specifies the high-level qualifier for the Language Environment (LE) runtime library. CEE should be set to the prefix of the IBM LE 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.

**Note:** Uncomment the SET CBC line and tailor the STEPLIB concatenation if these libraries are not in your system link library concatenation.

### Listing B-3 Sample JCL for crmlkoff Command

---

```
//*****
/* THIS JOB IS USED FOR THE STAND-ALONE LINK COMMAND          *
/* TO DEACTIVATE A REMOTE LINK. SEE USER GUIDE FOR MORE INFO  *
/*
/* @(#) $Id: crmlkoff.jcl,v 1.10 2001/05/07 23:41:27 crount Exp $ *
/* Copyright (c)2000 BEA Systems, Inc., all rights reserved.   *
/******
/* YOU MUST SET THE ENVIRONMENT VARIABLES NEEDED BY CRMLKOFF  *
/******
/* LINKCMD INDICATES THE DISTRIBUTED SNACRM ADDRESS AND LINKNAME*
/* OBJLIB IS THE LOAD LIBRARY CONTAINING THE ELINK ADAPTER FOR  *
/* MAINFRAME PROGRAM OBJECTS                                   *
/* RUNOPTS SETS ANY DESIRED LE RUNTIME OPTIONS                 *
/* DATA IS THE DATASET THAT CONTAINS THE ENVIRONMENT VARIABLES *
/* ENVFILE NAMES THE MEMBER THAT CONTAINS THE ENVIRONMENT VARS *
/* SIZE SETS THE REGION SIZE FOR THE SNACRM PROCESS           *
/* TAILOR YOUR JCL FOR THE BELOW IF THESE LIBRARIES ARE NOT    *
/* IN YOUR SYSTEM LINK LOAD LIBRARY CONCATENATION              *
/* 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 DATA=
/* SET ENVFILE=ENV
/* 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
//ENV        DD  DSN=&DATA (&ENVFILE) ,DISP=SHR
//MSGFILE    DD  SYSOUT=*
//SYSPRINT   DD  SYSOUT=*
//
```

---

## Sample JCL for the CRMDOWN Command

This section provides an explanation of the `SET` commands and an example of JCL that can be used when you run the `crmdown` command.

SET STOPCMD	Sets the <code>crmdown</code> command line parameters. Refer to “ <a href="#">crmdown</a> ” for more information about the command line parameters.
SET OBJLIB	Indicates the name of the PDSE library where the <code>crmdown</code> executable is installed.
SET DATA	Indicates the dataset containing the <code>ENVFILE</code> .
SET ENVFILE	Indicates the name of the PDS member that contains the environment variables for the <code>crmdown</code> . A sample member, <code>ENV</code> , is delivered with your product.
SET SIZE	Defines the region size for the running <code>crmdown</code> task.
SET ENV	Indicates the <code>ENVFILE</code> 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 <code>crmdown</code> .
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.

**Note:** Uncomment the `SET CBC` line and tailor the `STEPLIB` concatenation if these libraries are not in your system link library concatenation.

### Listing B-4 Sample JCL for CRMDOWN Command

---

```
//*****
/* THIS JOB IS USED FOR THE STAND-ALONE COMMAND USED *
/* TO SHUTDOWN THE SNACRM PROCESS. SEE USER GUIDE FOR MORE INFO *
/* *
/* @(#) $Id: crmdown.jcl,v 1.5 2001/05/07 23:41:27 crount Exp $ *
/* Copyright (c)2000 BEA Systems, Inc., all rights reserved. *
/******
/* YOU MUST SET THE ENVIRONMENT VARIABLES NEEDED BY CRMDOWN *
/******
/******
/* STOPCMD INDICATES THE COMMAND LINE FOR CRMDOWN *
/* OBJLIB IS THE LOAD LIBRARY CONTAINING THE PROGRAM EXECUTABLES *
/* RUNOPTS SETS ANY DESIRED LE RUNTIME OPTIONS (OPTIONAL) *
/* DATA IS THE DATASET THAT CONTAINS THE ENVIRONMENT VARIABLES *
/* ENVFILE NAMES THE MEMBER THAT CONTAINS THE ENVIRONMENT VARS *
/* SIZE SETS THE REGION SIZE FOR THE SNACRM PROCESS *
/* *
/* TAILOR YOUR JCL FOR THE BELOW IF THESE LIBRARIES ARE NOT *
/* IN YOUR SYSTEM LINK LOAD LIBRARY CONCATENATION *
/* CEE IS THE HLQ FOR THE LE RUNTIME LIBRARY *
/* CBC IS THE HLQ FOR THE C/C++ RUNTIME LIBRARY *
/******
/* SET STOPCMD='-n<host name>:<port>'
/* SET OBJLIB=
/* SET RUNOPTS=
/* SET DATA=
/* SET ENVFILE=ENV
/* 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/&STOPCMD'
//STEPLIB DD DSN=&OBJLIB,DISP=SHR
/* DD DSN=&CEE..SCEERUN,DISP=SHR
/* DD DSN=&CBC..SCLBDLL,DISP=SHR
//ENV DD DSN=&DATA(&ENVFILE),DISP=SHR
//MSGFILE DD SYSOUT=*
//SYSPRINT DD SYSOUT=*

//
```

---



# 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 LOGSCMD	Sets the CRMLOGS command line parameters. Refer to <a href="#">CRMLOGS</a> for more information about the command line parameters.
SET OBJLIB	Indicates the name of the PDSE library where the CRMLOGS executable is installed.
SET DATA	Indicates the dataset containing the ENVFILE.
SET ENVFILE	Indicates the dataset containing the ENVFILE.
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.

**Note:** Uncomment the SET CBC line and tailor the STEPLIB concatenation if these libraries are not in your system link library concatenation.

## Listing B-5 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 1.6 2001/05/07 23:41:27 crount Exp $ *
/* Copyright (c)2000 BEA Systems, Inc., all rights reserved. *
/*****
/* YOU MUST SET THE ENVIRONMENT VARIABLES NEEDED BY CRMLOGS *
/*****
/*****
/* SNACMD IS USED TO SET THE DESIRED SNACRM GROUP NAME      *
/* OBJLIB IS THE LOAD LIBRARY CONTAINING THE EAM PROGRAM OBJECTS*

```

## B Sample JCL for an MVS Platform

---

```
//* RUNOPTS SETS ANY DESIRED LE RUNTIME OPTIONS (OPTIONAL) *
//* DATA IS THE DATASET THAT CONTAINS THE ENVIRONMENT VARIABLES *
//* ENVFILE NAMES THE MEMBER THAT CONTAINS THE ENVIRONMENT VARS *
//* SIZE SETS THE REGION SIZE FOR THE SNACRM PROCESS *
//* ENV SETS THE ENVIRONMENT VARIABLES DD NAME *
//* TAILOR YOUR JCL FOR THE BELOW IF THESE LIBRARIES ARE NOT *
//* IN YOUR SYSTEM LINK LOAD LIBRARY CONCATENATION *
//* CEE IS THE HLQ FOR THE LE RUNTIME LIBRARY *
//* CBC IS THE HLQ FOR THE C/C++ RUNTIME LIBRARY *
/******
// SET LOGSCMD=<group>
// SET OBJLIB=
// SET DATA=
// SET ENVFILE=ENV
// 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/&LOGSCMD'
//STEPLIB DD DSN=&OBJLIB,DISP=SHR
//* DD DSN=&CEE..SCEERUN,DISP=SHR
//* DD DSN=&CBC..SCLBDLL,DISP=SHR
//MSGFILE DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//ENV DD DSN=&DATA(&ENVFILE),DISP=SHR
//
```

---

# C CRM Error Messages

The following Communications Resource Manager (CRM) messages can be displayed in the Tuxedo console window.

9001:ERROR	<taskname> timed out with failCode <failcode>
DESCRIPTION	<p>A conversation has timed out in the CRM with the stack return code of &lt;failcode&gt;. A timer event set to watch a conversation has expired.</p> <p>&lt;taskname&gt; may appear as:</p> <p>OB-Conversation #nn (&lt;linkref&gt;) tx #m &lt;tranname&gt;, or</p> <p>IB-Conversation #nn (&lt;linkref&gt;) tx #m &lt;tranname&gt;</p> <p>where:</p> <p>nn is an internal APPC conversation number and m is the transaction context where -1 signifies non-transactional.</p> <p>Outbound Conversation nnnn Link (&lt;linkref&gt;) TCTXT (tctxt) Tran(&lt;tranname&gt;), or</p> <p>Inbound Conversation nnnn Link (&lt;linkref&gt;) TCTXT (tctxt) Tran(&lt;tranname&gt;)</p> <p>where</p> <p>nnnn is the CRM Logical Task Number and tctxt is the transaction context, where FFFFFFFF indicates notn-transactional.</p>
ACTION	Examine stderr and the ULOG for additional information concerning the failure.
9002:ERROR	Unable to create APPC Server for %s.

	DESCRIPTION	CRM was unable to instantiate the stack object due to an error.
	ACTION	Check for additional messages in <code>stderr</code> . 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	<p>The &lt;returncode&gt; is the value returned by the SNA Stack software. Check the status of the stack, the configuration of the stack, and the gateway configuration.</p> <p><b>Note:</b> Unless you started the CRM as a Tuxedo server, you must manually kill the CRM process.</p>
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, TCTXT(%s), blob dropped. Transaction presumed forgotten.	
	DESCRIPTION	An attempt was made 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	<p>A conversation has failed with the stack return code of &lt;failcode&gt;.</p> <p>&lt;taskname&gt; may appear as:</p> <p>Outbound Conversation nnnn Link &lt;linkref&gt; TCTXT (&lt;tctxt&gt;) Tran &lt;tranname&gt;, or</p> <p>Inbound Conversation nnnn Link &lt;linkref&gt; TCTXT (&lt;tctxt&gt;) Tran &lt;tranname&gt;,</p> <p>where:</p> <p>nnnn is the CRM Logical TAsk Number.</p> <p>tctxt is the transaction context where FFFFFFFF signifies non-transactional.</p> <p>Possible values for the &lt;failcode&gt; are:</p> <ol style="list-style-type: none"> <li>1. Communications - unable to create the APPCserver object.</li> <li>2. MemoryAllocation - internal error allocating memory.</li> <li>3. InvalidObject - a CRM object could not be created or has been made invalid by some previous error.</li> <li>4. InputOutput - error occurred during file I/O or an unexpected APPC return code was received.</li> <li>5. Registration - internal task cannot be registered.</li> </ol>

	ACTION	Examine <code>stderr</code> 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 GWSNAX 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 <code>stackref</code> in the link configuration is incorrect.
	ACTION	Correct the <code>stackref</code> that is in error, run <code>dmloadcf</code> , 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	<p>An internal state table failure has occurred. The &lt;additional information&gt; will be one of the following:</p> <ol style="list-style-type: none"> <li>1. From &lt;oldstate&gt; to &lt;newstate&gt; for &lt;dir&gt;-bound transaction TCTXT&lt;tid&gt;</li> <li>2. To &lt;newstate&gt; for inbound transaction TCTXT&lt;tid&gt;</li> <li>3. To &lt;newstate&gt; for outbound transaction TCTXT&lt;tid&gt;</li> </ol>
	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 = <tctx>	
	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, TCTXT(%s)	
	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 GWSNAX and CRM. Possibly incompatible eLink Adapter for Mainframe 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:ERROR	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:ERROR	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: <ol style="list-style-type: none"> <li>1. Sync-Level, function, and API</li> <li>2. Function and API</li> </ol>
	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, <member>	
	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	CRM encryption setup failed	
	DESCRIPTION	An error has occurred while establishing link-level encryption with the CRM.
	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 CRM client	
	DESCRIPTION	A client has attempted to access the CRM without the proper authentication or encryption setup.
	ACTION	Verify that the client should have access to the CRM. Verify that encryption is set up correctly in both the CRM 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	CRM authentication setup failed	
	DESCRIPTION	An error has occurred while authenticating a CRM client.
	ACTION	Verify that the authentication file(s) for both the CRM 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 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 particular 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 LUs.

### **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 gateway 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**

### **T 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.

### **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

## A

Advanced Peer-to-Peer Networking (APPN)  
1-6

## C

CICS/ESA LU

configuration 1-15

commands

CRMDOWN A-8

CRMLKOFF A-7

CRMLKON A-6

CRMLOGS A-5

SET CEE 2-7, B-2

SET OBJLIB 2-7, B-2

SNACRM A-1

Communications Resource Manager 1-5

communications resource manager

about 1-2

configuration

CICS/ESA LU 1-15

distributed 1-4

JAM gateway for Virtual

Telecommunications Access

Method 1-4

local 1-3

system 1-3

CRM

starting on OS/390 MVS 2-6

starting on UNIX or Windows NT 2-2

CRM on OS/390

stopping 2-23

CRM. See Communications Resource  
Manager

CRMDOWN commands

SET CBC 2-25, B-7

SET CEE 2-25, B-7

SET DATA2 2-25, B-7

SET ENVFILE 2-25, B-7

SET LNKCMD 2-25, B-7

SET OBJLIB 2-25, B-7

SET SIZE 2-25, B-7

crmlkoff command 2-2, 2-17, 2-23, A-7

CRMLKOFF commands

SET CBC 2-19, B-6

SET CEE 2-19, B-6

SET DATA2 2-18, B-5

SET ENV 2-19, B-5

SET ENVFILE 2-18, B-5

SET LINKCMD 2-18, B-5

SET OBJLIB 2-18, B-5

SET SIZE 2-19, B-5

crmlkon command 2-13, A-6

CRMLKON commands

SET CBC 2-15, B-4

SET CEE 2-15, B-4

SET DATA2 2-15, B-4

SET ENV 2-15, 2-25, B-4, B-7

SET ENVFILE 2-15, B-4

SET LNKCMD 2-15, B-3

SET OBJLIB 2-15, B-4

SET SIZE 2-15, B-4

---

## CRMLOGS command

- SET CEE 2-21, B-9
- SET ENV 2-21, B-9
- SET SIZE 2-21, B-9

## crmlogs command A-5

## CRMLOGS commands

- SET CBC 2-21, B-9
- SET DATA1 2-21, B-9
- SET DATA2 2-21
- SET ENVFILE 2-21, B-9
- SET LNKCMD 2-21, B-9
- SET OBJLIB 2-21, B-9

customer support contact information x

## D

documentation, where to find it viii

## E

error messages C-1

## J

jsnacrm utility 2-9

## L

link

- start command 2-13
- stop command 2-17, A-7

links

- activating and de-activating 2-12

Local Area Network (LAN) 1-6

log files

- SNACRM A-5

## P

protocols 1-2

PU2.1 server (stack) 1-3

## R

related information viii

## S

servers

- SNACRM A-1

SET SNACMD command 2-6, B-2

SNA protocol 1-3

SNA stack

- third-party 1-3, 1-4

SNACRM

- command A-1

SNACRM commands

- SET CBC 2-7, B-2
- SET DATA2 2-7, B-2
- SET ENV 2-7, B-2
- SET ENVFILE 2-7, B-2
- SET OBJLIB 2-7, B-2
- SET SIZE 2-7, B-2
- SET STARTCMD 2-6, B-2

specifying

- Virtual Telecommunications Access  
Method 1-4

support

- technical x

system configuration 1-3

## T

TCP/IP 1-5

## V

Virtual Telecommunications Access Method  
configuration

- JAM gateway 1-4
- specifying 1-4

VTAM. See Virtual Telecommunications  
Access Method



---

## **W**

winner sessions 1-15

