

# **Oracle® Tuxedo Mainframe Adapter for SNA**

CRM Administration Guide

12c Release 2 (12.2.2)

August 2016

**ORACLE®**

Copyright © 2007, 2016, Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

This software or hardware and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

# Contents

## Introduction

Audience .....	1-2
Related Information .....	1-2
Documentation Conventions .....	1-2

## Understanding the Communications Resource Manager

About the Communications Resource Manager .....	2-1
System Configuration .....	2-2
Prerequisite for Mainframe Knowledge .....	2-2
Local Configuration .....	2-3
Distributed Configurations .....	2-3
Preparing Mainframe Configurations for CRM Requirements .....	2-5
Requirements for Establishing a VTAM Configuration .....	2-6
Configuring the CRM for a Third-Party Stack .....	2-6
Third-Party Stack Configuration .....	2-6
VTAM Cross Platform Definitions .....	2-13
Configuring the CRM for the z/OS Platform .....	2-15
Activating the APPLID for CRM .....	2-16
Configuring the CICS/ESA LU .....	2-16
Putting It All Together .....	2-18
Example of Windows or UNIX Cross-Platform Definitions .....	2-18
Example of z/OS Definitions .....	2-22

# Administering the SNA Components

Starting the CRM . . . . .	3-1
Starting the CRM on UNIX . . . . .	3-2
CRM Command . . . . .	3-2
Command Line Options . . . . .	3-3
Examples . . . . .	3-4
Diagnostics . . . . .	3-5
Starting the CRM on z/OS MVS . . . . .	3-5
Sample JCL for the CRM Command . . . . .	3-6
Having Separate CRM Configurations on Mainframe . . . . .	3-8
Configurations on CRM Side . . . . .	3-8
Configurations on TMA SNA Side . . . . .	3-13
Using the CRM Monitor . . . . .	3-13
Launching the CRM Monitor from the Windows Desktop . . . . .	3-13
Launching the CRM Monitor from the Command Line . . . . .	3-13
Setting CRM Monitor Options . . . . .	3-14
Multiple SNA Gateways Connecting to a Single CRM . . . . .	3-16
Overview . . . . .	3-16
Advantages of Multi-Gateway Support . . . . .	3-16
Multi-Gateway Connection . . . . .	3-17
Multi-Gateway Shutdown . . . . .	3-17
Load Balancing for Inbound Request . . . . .	3-18
Transaction Affinity . . . . .	3-18
Activating and De-Activating Links . . . . .	3-18
crmlkon Command . . . . .	3-19
Command Line Options . . . . .	3-19
Example . . . . .	3-20

Diagnostics . . . . .	3-20
Sample JCL for the crmlkon Command . . . . .	3-20
crmlkoff Command . . . . .	3-22
Command Line Options . . . . .	3-23
Example. . . . .	3-24
Diagnostics . . . . .	3-24
Sample JCL for the crmlkoff Command. . . . .	3-24
Reviewing CRM Log Files . . . . .	3-26
CRMLOGS Command . . . . .	3-27
Command Line Options . . . . .	3-27
Example. . . . .	3-27
Diagnostics . . . . .	3-27
Sample JCL for the CRMLOGS Command . . . . .	3-27
Stopping the CRM. . . . .	3-29
crmdown Command . . . . .	3-30
Command Line Options . . . . .	3-30
Example. . . . .	3-31
Diagnostics . . . . .	3-31
Sample JCL for the CRMDOWN Command. . . . .	3-31

## Command Reference Pages

CRM . . . . .	A-1
CRMLOGS . . . . .	A-6
crmretc . . . . .	A-7
crmlkon . . . . .	A-8
crmlkoff . . . . .	A-9
crmdown . . . . .	A-10

## Sample JCL for an MVS Platform

Sample JCL for the CRM Command . . . . .	B-1
Sample JCL for the crmlkon Command . . . . .	B-4
Sample JCL for the crmlkoff Command . . . . .	B-6
Sample JCL for the CRMDOWN Command . . . . .	B-8
Sample JCL for the CRMLOGS Command . . . . .	B-10

## CRM Error Messages

## Glossary

# Introduction

This document explains how the Oracle Tuxedo Mainframe Adapter for SNA 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”](#)

# Audience

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.

# Related Information

The following documents contain information that is relevant to using Oracle Tuxedo Mainframe Adapter for SNA:

- *Oracle Tuxedo Mainframe Adapter for SNA Release Notes*
- *Oracle Tuxedo Mainframe Adapter for SNA Installation Guide*
- *CRM Administration Guide*
- *User Guide*
- *Oracle Tuxedo Mainframe Adapter for SNA 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.
<i>italics</i>	Indicates emphasis or book titles.



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

Convention	Item
...	<p data-bbox="341 361 821 378">Indicates one of the following in a command line:</p> <ul data-bbox="341 395 1053 493" style="list-style-type: none"> <li data-bbox="341 395 1020 413">• That an argument can be repeated several times in a command line</li> <li data-bbox="341 430 908 447">• That the statement omits additional optional arguments</li> <li data-bbox="341 465 1053 482">• That you can enter additional parameters, values, or other information</li> </ul> <p data-bbox="341 510 729 527">The ellipsis itself should never be typed.</p> <p data-bbox="341 545 431 562"><i>Example:</i></p> <pre data-bbox="341 579 1002 635">buildobjclient [-v] [-o name ] [-f file-list]... [-l file-list]...</pre>
. . .	<p data-bbox="341 670 1053 687">Indicates the omission of items from a code example or from a syntax line.</p> <p data-bbox="341 696 807 713">The vertical ellipsis itself should never be typed.</p>

# 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 Oracle Tuxedo Mainframe Adapter for SNA 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 OracleTuxedo Mainframe Adapter for SNA 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 OracleTuxedo Mainframe Adapter for SNA Gateway. For a complete list of supported operating systems, refer to the Oracle Tuxedo Mainframe Adapter for SNA *Release Notes*.

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

## System Configuration

The OracleTuxedo Mainframe Adapter for SNA 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 Oracle Tuxedo Mainframe Adapter for SNA *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 OracleTuxedo Mainframe Adapter for SNA, HP-UX 11.23 on PA-RISC is the only non-mainframe platform for which the CRM is available.

## Prerequisite for Mainframe Knowledge

Before you begin configuring CRM on Mainframe, it is important to take the following items into consideration:

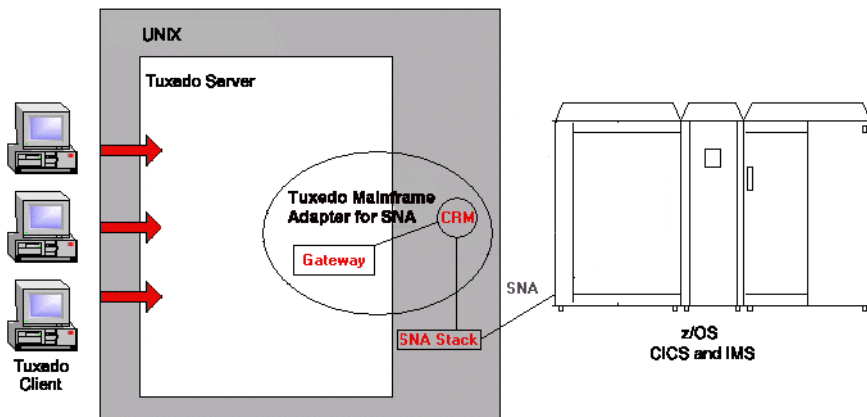
- Basic knowledge about Mainframe, such as allocated datasets, submit JCL, and such z/OS operations.
- Familiarity with the basic VTAM concepts, such as VTAM Major nodes, Logic Unit, etc.
- Familiarity with CICS, including defining CICS resources, such as connections, sessions, transactions, programs, etc.

- Familiarity with IMS, if you need to access the IMS system, including defining IMS used resources, such as IMS PSB, ACB, APPC/IMS, etc.

## Local Configuration

The local configuration combines the OracleTuxedo Mainframe Adapter for SNA Gateway and the CRM with the stack (PU2.1 server) on the same UNIX machine with the CRM configured to run as a Tuxedo server rather than a standalone process. It employs the IBM proprietary SNA protocol for transactions with the mainframe via the stack.

Figure 2-1 OracleTuxedo Mainframe Adapter for SNA Local Configuration

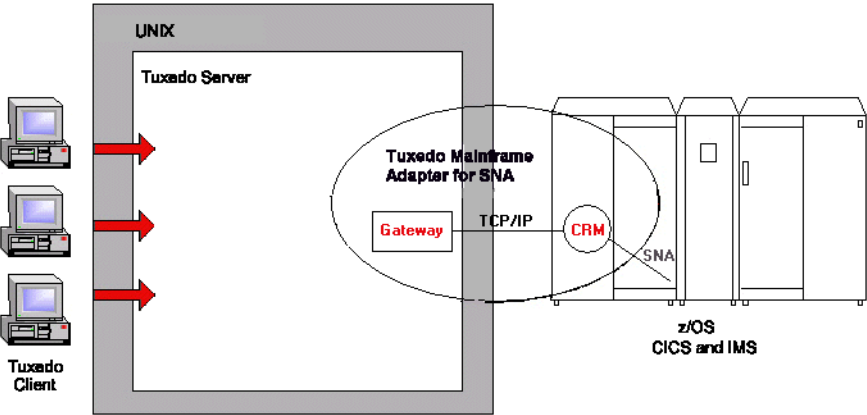


## Distributed Configurations

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 OracleTuxedo

Mainframe Adapter for SNA Gateway configuration file. [Figure 2-2](#) illustrates the distributed configuration architecture.

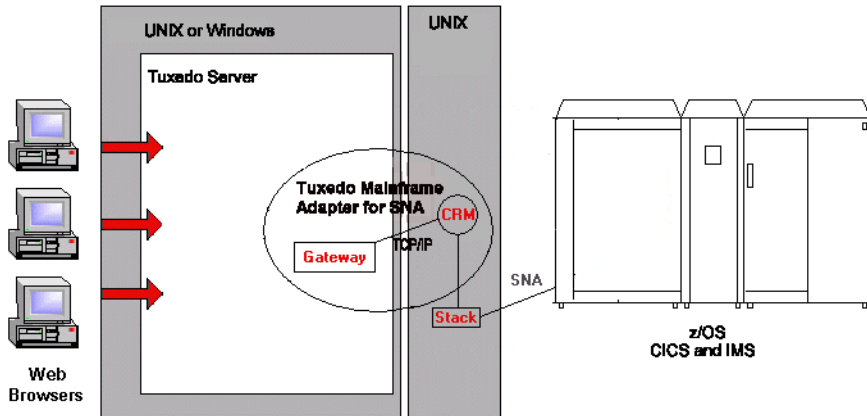
**Figure 2-2 OracleTuxedo Mainframe Adapter for SNA Distributed Configuration**



Another type of distributed configuration separates the CRM from the OracleTuxedo Mainframe Adapter for SNA Gateway on different UNIX or Windows machines. It employs Transmission Control Protocol/Internet Protocol (TCP/IP) connectivity between the OracleTuxedo Mainframe Adapter for SNA 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 OracleTuxedo Mainframe Adapter for SNA Gateway and the remote CRM.

**Figure 2-3 Non-Mainframe OracleTuxedo Mainframe Adapter for SNA 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 OracleTuxedo Mainframe Adapter for SNA 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. VTAM is the IBM subsystem that implements Systems Network Architecture (SNA) for mainframe environments. VTAM provides an Application Programming Interface (API) for communication applications and controls communication equipment such as adapters and controllers. In modern terminology, VTAM provides a communication stack and device drivers. VTAM performs a number of tasks in a network. For example, VTAM:

- Monitors and controls the activation and connection of resources
- Establishes connections and manages the flow and pacing of sessions

- Provides API (for example, an APPC API for LU 6.2 programming) that allow access to the network by user-written application programs and IBM-provided subsystems
- Provides interactive terminal support for Time Sharing Option (TSO) using Multiple Virtual Storage (MVS)
- Provides support for both locally and remotely attached resources

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 OracleTuxedo Mainframe Adapter for SNA 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 SNAplus2 stack for HP-UX 11.23.

### 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 2-1](#) 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 OracleTuxedo Mainframe Adapter for SNA configuration is defined.

### Listing 2-1 SNA Node Configuration File

---

```
[define_node_config_file]
major_version = 5
minor_version = 1
update_release = 1
revision_level = 109
olar_supported = YES

[define_node]
cp_alias = LCHP2
description = ""
fqcp_name = BEALAN.LCHP2
node_type = END_NODE
mode_to_cos_map_supp = NO
mds_supported = YES
node_id = <03010300>
max_locates = 100
dir_cache_size = 255
max_dir_entries = 0
locate_timeout = 60
reg_with_nn = YES
reg_with_cds = YES
mds_send_alert_q_size = 100
cos_cache_size = 24
tree_cache_size = 40
tree_cache_use_limit = 40
max_tdm_nodes = 0
max_tdm_tgs = 0
max_isr_sessions = 1000
isr_sessions_upper_threshold = 900
isr_sessions_lower_threshold = 800
```

```
isr_max_ru_size = 16384
isr_rcv_pac_window = 8
store_endpt_rscvs = NO
store_isr_rscvs = NO
store_dlur_rscvs = NO
dlur_support = YES
pu_conc_support = NO
nn_rar = 128
max_ls_exception_events = 0
ptf_flags = NONE
cplu_syncpt_support = NO
```

```
[define_ethernet_dlc]
dlc_name = ETHER1
description = ""
neg_ls_supp = YES
initially_active = NO
adapter_number = 0
```

```
[define_ethernet_port]
port_name = ETSAP1
description = ""
dlc_name = ETHER1
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 = <00306E4B25F0>
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
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
test_timeout = 5
test_retry_limit = 2
xid_timeout = 5
xid_retry_limit = 2
t1_timeout = 5
t1_retry_limit = 5

```

```

[define_ethernet_ls]
ls_name = ZOS4HP2
description = Link to lczos4
port_name = ETSAP1
adj_cp_name = ADCD.ADCDSSCP
adj_cp_type = LEARN_NODE
mac_address = <0002B3C008C8>
lsap_address = 0x04
auto_act_supp = NO
tg_number = 0
limited_resource = NO
solicit_sscp_sessions = YES
pu_name = <0000000000000000>
disable_remote_act = NO

```

```
default_nn_server = NO
dspu_services = NONE
dspu_name = <0000000000000000>
dlus_name = <0000000000000000000000000000000000>
bkup_dlus_name = <0000000000000000000000000000000000>
hpr_supported = NO
hpr_link_lvl_error = NO
link_deact_timer = 30
use_default_tg_chars = YES
ls_attributes = SNA
adj_node_id = <03010300>
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 = 0
user_def_parm_2 = 0
user_def_parm_3 = 0
target_pacing_count = 7
max_send_btu_size = 1033
ls_role = USE_PORT_DEFAULTS
max_ifrm_rcvd = 0
dlus_retry_timeout = 0
dlus_retry_limit = 0
initially_active = NO
restart_on_normal_deact = NO
react_timer = 30
react_timer_retry = 65535
test_timeout = 5
test_retry_limit = 2
xid_timeout = 5
xid_retry_limit = 2
t1_timeout = 5
t1_retry_limit = 5
```

```

[define_local_lu]
lu_alias = Z4HP2A
list_name = ""
description = ""
lu_name = Z4HP2A
lu_session_limit = 20
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_partner_lu]
plu_alias = ""
description = (Auto defined - default LU)
fqplu_name = ADCD.LCZOS4
plu_un_name = <0000000000000000>
parallel_sess_supp = YES
max_mc_ll_send_size = 0
conv_security_ver = NO

[define_partner_lu]
plu_alias = TMACICS
description = ""
fqplu_name = ADCD.TMACICS
plu_un_name = TMACICS
parallel_sess_supp = YES
max_mc_ll_send_size = 0
conv_security_ver = NO

[define_mode]
mode_name = SMSNA100
description = ""

```

```

max_neg_sess_lim = 256
plu_mode_session_limit = 10
min_conwin_src = 5
min_conloser_src = 0
auto_act = 0
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 = ADCD.LCZOS4
resource_type = ENCP_RESOURCE
description = ""
parent_name = <00000000000000000000000000000000>
parent_type = ENCP_RESOURCE

[define_directory_entry]
resource_name = ADCD.LCZOS4
resource_type = LU_RESOURCE
description = (Auto defined - default LU)
parent_name = ADCD.LCZOS4
parent_type = ENCP_RESOURCE

[define_directory_entry]
resource_name = ADCD.ADCDSSCP
resource_type = ENCP_RESOURCE
description = ""
parent_name = <00000000000000000000000000000000>
parent_type = ENCP_RESOURCE

[define_directory_entry]
resource_name = ADCD.TMACICS
resource_type = LU_RESOURCE
description = ""
parent_name = ADCD.ADCDSSCP

```

```
parent_type = ENCP_RESOURCE

[define_directory_entry]
resource_name = ADCD.ADCDSSCP
resource_type = LU_RESOURCE
description = (Auto defined - default LU)
parent_name = ADCD.ADCDSSCP
parent_type = ENCP_RESOURCE
```

---

## 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

The external communication adapter (XCA) major node defines VTAM's connections to the following shared access transport facilities (SATFs):

- Local area networks (LANs)
- Asynchronous transfer mode (ATM) networks accessed through:
  - LAN emulation
  - Native ATM
- Enterprise Extender (EE) networks

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 2-2 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 2-3](#)) 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 2-3 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 the z/OS Platform

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

The APPLID definition shown in [Listing 2-4](#) shows an example of a VTAM stack configuration to run the CRM on z/OS.

### Listing 2-4 APPLID Definition (z/OS)

```

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

```

---

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

#### ACBNAME

Specifies the minor node name assigned to this application program. This name must be unique within the domain in which the application program resides.

SYNCLVL

Specifies the application program synchronization support level.

SYNCPT

Specifies that synchronization levels of SYNCPT, CONFIRM, and NONE are supported. This logical unit supports both conversation-specified confirmation and sync point processing.

PARSESS

Allows this application program to have multiple LU-LU sessions between the same session partners.

## Activating the APPLID for CRM

Use the following MVS command to activate APPLID for CRM dynamically:

```
V NET, ID=BEASNA, ACT, SCOPE=ALL
```

## 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 OracleTuxedo Mainframe Adapter for SNA 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)
```

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

AUTOCONNECT

For systems using ACCESSMETHOD(VTAM), you specify with AUTOCONNECT(YES) or (ALL) that sessions are to be established (that is, BIND is to be performed). Such sessions are set up during CICS initialization, or when you use the CEMT or EXEC CICS SET VTAM OPEN command to start communication with VTAM. If the connection cannot be made at these times when the remote system is unavailable, you must subsequently acquire the link

by using the CEMT or EXEC CICS SET CONNECTION(sysid) INSERVICE ACQUIRED command, unless the remote system becomes available in the meantime and initiates communications.

For APPC connections with SINGLESESS(NO) specified, CICS tries to bind, on system start-up, the LU services manager sessions in mode group SNASVCMG.

AUTOCONNECT(NO) CICS does not attempt to bind sessions when the connection is established.

ACCESSMETHOD(VTAM)

VTAM Communication between the local CICS region and the system defined by this connection definition is through VTAM. You can use VTAM intersystem communication (ISC) for systems that are in different MVS images or in different address spaces in the same MVS image.

NETNAME

Specifies the network name that identifies the remote system. The name can be up to eight characters in length. The name follows assembler language rules. It must start with an alphabetic character. Here, NETNAME is the APPLID for CRM.

## 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 OracleTuxedo Mainframe Adapter for SNA session definition:

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

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

AUTOCONNECT

Indicates how the activation of the session is negotiated.

YES

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

MODENAME

Indicates either a CICS/ESA-supplied mode name, such as SMSNA100, or your own defined mode name. If another set of session definitions exists for the BEA connection,

this mode name must be unique among all sets defined to the connection. The mode name corresponds to the VTAM LOGMODE name. A mode name defines characteristics such as pacing level and class of service. When transaction programs request a conversation, they usually do not specify which session to use for the conversation, but they can specify the mode name of the session. Each mode name is defined in the logon mode table, which describes the session parameters.

#### 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 OracleTuxedo Mainframe Adapter for SNA software, review [Table 2-1](#) 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 2-1](#) is the HP-UX SNAplus2 stack.

[Listing 2-5](#) shows the corresponding OracleTuxedo Mainframe Adapter for SNA Gateway configuration file. Refer to the *Oracle OracleTuxedo Mainframe Adapter for SNA User Guide* for information about setting up the Gateway configuration.

**Table 2-1 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> . CICCSSYN
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= <b>SNANET1</b> . SPARC cp_alias= <b>SPARC1</b>
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> "

**Table 2-1 Summary of DCL Definitions**

Name	Originates In	Needed In
CICS LU Name (e.g. <b>CICSSYN</b> )	VTAM-LU definition	<p><b>DCL-based Stack Configuration:</b>            Example:            fqplu_name=SNANET1.CICSSYN            plu_alias=CICSSYN</p> <p><b>GWSNAX Configuration:</b>            Example:            DM_SNALINKS            RLUNAME="CICSSYN"</p>
Terminal Identifier (e.g. <b>05DF0024</b> )	VTAM (IDNUM+IDBLK)	<p><b>DCL-based Stack Configuration:</b>            Example:            node_id=&lt;05000002&gt;</p>
SYNCLVL	Stacks	<p><b>DCL-based Stack Configuration:</b>            Example:            [define local_lu]            syncpt_support=CONFIRM</p> <p><b>GWSNAX Configuration:</b>            Example:            DM_SNALINKS            MAXSYNCLVL=1</p>
Map all incoming conversations to OracleTuxedo Mainframe Adapter for SNA Gateway (make sure TPs have all privileges available, e.g. CNOS, service conversations, etc.).	Stacks	<p><b>DCL-based Stack Configuration:</b>            Example:            Sna_tps            &lt;404040...hex representation of 64            EBCDIC spaces...404040&gt;            TYPE=QUEUED            TIMEOUT=-1            USERID=authorized_user_here            GROUP=authorized_group_here            LUALIAS=LOF0024A</p> <p><b>GWSNAX Configuration:</b>            Example:            DM_LOCAL_SERVICES            RNAME="DPL1SVR"</p>
CICS DPL program name (e.g. TOUPPER)	CICS/ESA	<p><b>GWSNAX Configuration:</b>            Example:            DM_REMOTE_SERVICES            RNAME=TOUPPER</p>

**Listing 2-5 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"

*DM_SNASTACKS

simpstk      SNACRM="simpcrm"
            STACKTYPE="HP62"
            LOCALLU="LOF0024A"
            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 Definitions

Before installing OracleTuxedo Mainframe Adapter for SNA software, review [Table 2-2](#) for an example of SNA definitions when the CRM runs on the z/OS platform. 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 2-6](#) shows the corresponding OracleTuxedo Mainframe Adapter for SNA configuration file. Refer to the *Oracle OracleTuxedo Mainframe Adapter for SNA User Guide* for information about setting up the Gateway configuration.



Table 2-2 Summary of z/OS SNA Definitions

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

**Table 2-2 Summary of z/OS SNA Definitions**

<b>Name</b>	<b>Originates In</b>	<b>Needed In</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 2-6 Corresponding DMCONFIG File**

```

#####

*DM_LOCAL_DOMAINS
simpsnad      GWGRP=GROUP2
              TYPE=SNAX
              DOMAINID="simpsnad"
              BLOB_SHM_SIZE=100000
              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

---

# Administering the SNA Components

The topics in this section cover activities an administrator performs with the Communications Resource Manager (CRM) to maintain Oracle Tuxedo Mainframe Adapter for SNA 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](#)
- [Having Separate CRM Configurations on Mainframe](#)
- [Using the CRM Monitor](#)
- [Multiple SNA Gateways Connecting to a Single CRM](#)
- [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 Oracle Tuxedo Mainframe Adapter for SNA 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 MVS system.
- Using the Tuxedo startup command, `tmbboot` 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 HP-UX 11.23 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 Oracle Tuxedo Mainframe Adapter for SNA 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:

**Table 3-1 CRM Command Options**

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>
-s	Off (if parameter is not used)	Optional	<p>Indicates APPC Stack API trace is turned on</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>

**Table 3-1 CRM Command Options**

<b>Keyword</b>	<b>Default</b>	<b>Required/ Optional</b>	<b>Description</b>
<code>-p &lt;nbr&gt;</code>	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>
<code>&lt;addr&gt;</code>	None	Required	<p>Specifies a TCP/IP address using <code>//hostname:port_addr</code> or the <code>sockaddr_in</code> format of family, port, address:</p> <p><code>&lt;0xFFFFPPPPAAAAAAAA&gt;</code></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>AAAAAAAA is the hex value of the IP address for the machine running the CRM.</p>
<code>&lt;group&gt;</code>	None	Required	<p>Indicates the Oracle Tuxedo Mainframe Adapter for SNA Gateway Group Name.</p>

## 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 Oracle Tuxedo Mainframe Adapter for SNA 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 MVS

The z/OS 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 MVS platform, it does not have to be restarted if the Oracle Tuxedo Mainframe Adapter for SNA 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 MVS version and the z/OS 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 3-1 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
//
```

---

## Having Separate CRM Configurations on Mainframe

TMA provides the flexibility for you to store and manage CRM related configurations (CICSLINKS and SNASTACKS) independently on mainframe.

More specifically, TMA automatically checks if `DM_SNASTACKS` is configured in `DMCONFIG`. If positive, TMA will use `DM_SNASTACKS` and `DM_SNALINKS` configurations in `DMCONFIG`; otherwise, TMA will use the CRM configurations on mainframe. If there are no CRM configurations on mainframe, TMA will be shut down with errors in `ULOG`.

To use the flexibility, do the following steps on both CRM side and TMA SNA side.

- [Configurations on CRM Side](#)
- [Configurations on TMA SNA Side](#)

After all configurations on both CRM side and TMA SNA side, start CRM firstly, then TMA SNA Gateway. You can use `ULOG` to check whether TMA SNA Gateway successfully connects to CRM.

## Configurations on CRM Side

On this side, do the followings.

1. For each CRM, define its configuration name and location by adding command lines like below.

```
...
// SET STARTCMD='-t 3 "//zsvr:8001" CRM8001'
//CFG DD DSN=&DATA(&CFGFILE),DISP=SHR
```

```
//SET CFGFILE=CFG
```

```
...
```

CFG is the DD card, specifying separate CRM configurations on mainframe.

SET CFGFILE indicates the name of the PDS member that contains STACK and LINK configurations on CRM.

2. Create dataset member (CFG) for each CRM, and set LINK and STACK parameters. [Table 3-2](#) lists all available parameters for CRM configuration; [Listing 3-2](#) is a sample for CRM configuration.

**Table 3-2 Parameters for CRM Configuration**

Section	Parameter	Notes
STACK	STACKTYPE=VT210	The same as STACKTYPE in *DM_SNASTACKS in DMCONFIG; however, its value can only be VT210.
	LOCALLU	The same as LOCALLU in *DM_SNASTACKS in DMCONFIG.
	LTPNAME	The same as LTPNAME in *DM_SNASTACKS in DMCONFIG.
	STACKPARMS	The same as STACKPARMS in *DM_SNASTACKS in DMCONFIG.

**Table 3-2 Parameters for CRM Configuration**

LINK	RDOM	The same as RDOM in *DM_SNALINKS in DMCONFIG. The value of RDOM must match the domain name of DM_REMOTE in DMCONFIG, so that SNAGW can map configurations.
	LSYSID	The same as LSYSID in *DM_SNALINKS in DMCONFIG.
	RSYSID	The same as RSYSID in *DM_SNALINKS in DMCONFIG.
	RLUNAME	The same as RLUNAME in *DM_SNALINKS in DMCONFIG.
	MODENAME	The same as MODENAME in *DM_SNALINKS in DMCONFIG.
	STARTTYPE	The same as STARTTYPE in *DM_SNALINKS in DMCONFIG.
	MAXSESS	The same as MAXSESS in *DM_SNALINKS in DMCONFIG.
	MINWIN	The same as MINWIN in *DM_SNALINKS in DMCONFIG.
	MAXSYNCLVL	The same as MAXSYNCLVL in *DM_SNALINKS in DMCONFIG.

**Listing 3-2 CRM Configuration Sample**

```
...  
[STACK]  
STACKTYPE=VT210  
LOCALLU=CRMLU05  
LTPNAME=*  
STACKPARMS=wasa  
[LINK]
```

```

RDOM=MVSDOM
LSYSID=CR05
RSYSID=CICS
RLUNAME=CICSA
MODENAME=SMSNA100
SECURITY=LOCAL
STARTTYPE=COLD
MAXSESS=8
MINWIN=4
MAXSYNCLVL=2
. . .

```

---

Below is a JCL sample for CRM command (separate CRM configurations on mainframe).

**Listing 3-3 JCL Sample for CRM Command (Separate CRM Configurations on Mainframe)**

---

```

/*****
/* 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 CFGFILE=CFG
// 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
//CFG      DD DSN=&DATA(&CFGFILE),DISP=SHR
//
```

---

## Configurations on TMA SNA Side

On this side, configure TMA SNA gateway according to [Configuring the Oracle Tuxedo Mainframe Adapter for SNA Gateway](#), except for its [Add the \\*DM\\_SNASTACKS Section](#) and [Add the \\*DM\\_SNALINKS Section](#) in [Option A: Edit the DMCONFIG File](#).

## 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 Oracle Tuxedo Mainframe Adapter for SNA.

To launch the CRM Monitor from the Desktop:

- Choose Programs | BEA Weblogic E-Business Platform | Oracle Tuxedo Mainframe Adapter for SNA 10g R3 | 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 JRE or JDK interpreter:

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

## Setting CRM Monitor Options

Figure 3-1 shows an example of the CRM Monitor display.

Figure 3-1 CRM Monitor Display

The screenshot shows the CRM Monitor application window with the following components and callouts:

- CRM Address:** Enter the address of the CRM to be monitored. (Callout pointing to the "Enter CRM Address" field containing `//dalvs5:8002`)
- Selected CRM:** Displays name of the CRM at the address. (Callout pointing to the "Address" field containing `//dalvs5:8002`)
- Trace Status:** Displays the currently selected trace option(s). (Callout pointing to the "Trace Status" field containing `Current CRM trace level is 0, Appc stack trace is OFF`)
- Trace Options:** Select the trace option(s). One of four CRM options and/or one of two APPC options. (Callout pointing to the "Trace" and "APPC Trace" buttons)
- Link Status:** Displays the current status of all remote links for the selected CRM. (Text may be scrolled.) (Callout pointing to the "Link Status" field containing `Remote Links: 00 WEBL<->CICS SMSNA100 Ins Acq Max=10 Active= 5 InUse= 4 A06CICS 000 001 002 004`)
- Link Statistics:** Displays the current statistics for all remote links for the selected CRM. (Text may be scrolled.) (Callout pointing to the "Link Statistics" field containing `Statistics for Remote Link A06CICS Input conversations = 0, messages = 5902, bytes = Output conversations = 5906, messages = 5905, bytes = 1.`)
- Messages:** Displays results of automatic connection attempts and/or entered commands. (Callout pointing to the "Signon to //dalvs5:8002 successful" message)

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 Oracle 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.

## Multiple SNA Gateways Connecting to a Single CRM

### Overview

TMA CRM permits multiple Gateways to connect and use the resources provided by a single CRM process. This feature considerably reduces the processing load required on the computer hosting the CRM while supporting distribution of workload at SNA Gateway level.

The multiple Gateways provide mainframe to Tuxedo application load balancing and failover.

### Advantages of Multi-Gateway Support

The primary advantage of Multi-Gateway support is the ability for the CRM to host concurrent connections from one or more SNA Gateways. The advantages of this approach include:

- Overall, a single CRM consumes less system resources than multiple CRMs.

- The resulting configuration is much easier for you to configure, administer, and operate.
- The CRM can load balance mainframe requests for services provided by the connected Tuxedo application.

## Multi-Gateway Connection

The CRM offers a single point of contact - that is, a single listener socket (identified by host IP address and port number) to which components (such as SNA Gateways) can address their connection requests. Each new connection request is handed off to a server socket dedicated to servicing that component connection.

The CRM attaches significance to the following component connect/disconnect scenarios:

- First Tuxedo SNA Gateway Connection

The first (or only) Gateway to connect to the CRM, the CRM accepts configuration information from the GWSNAX to initialize the VTAM interface and to establish links with mainframe systems, such as CICS and IMS.

When CRM configuration is complete, the GWSNAX receives a response indicating that the CRM is ready to begin processing requests.

- Subsequent Tuxedo SNA Gateway Connections

If the client is a Tuxedo SNA Gateway and there are other Gateways currently connected to the CRM, the GWSNAX bypass the configuration upload operation. In this case, the first Tuxedo SNA Gateway client has already supplied the CRM configuration information and configuration information is not required.

## Multi-Gateway Shutdown

Oracle Tuxedo SNA gateway provides support for the following orderly disconnection and shutdown procedures.

- Shutdown Processing (All Gateways Except the Last)

During shutdown, a Gateway disconnects (closes the socket connection), Phase 1 shutdown processing is initiated for that connection only. Phase 1 shutdown is a "shutdown pending" state in which the following is true:

- Sends and receives in progress are permitted to complete.
- In-flight transactions are aborted.

The CRM will not proceed to Phase 2 shutdown processing if other Gateways are still connected. Links to mainframe systems will remain active and the CRM will continue normal processing for other connected Gateways.

- Shutdown Processing (Last Gateway)

When the last (or only) Gateway requests shutdown or closes its connection with the CRM, the CRM executes Phase 1 shutdown processing for that Gateway as described in Shutdown Processing (All Gateways Except the Last). However, when Phase 1 shutdown processing is complete, the CRM proceeds to Phase 2 shutdown processing, in which the following occurs:

- Links to mainframe systems are stopped.
- The current configuration is discarded.
- The CRM returns to a "reset" state pending connection by another client.

## Load Balancing for Inbound Request

TMA SNA adapter takes advantage of the load balancing features to dispatch a request from mainframe to current active SNA gateways that are connecting to CRM. The request from mainframe to GWSNAX is always load balanced via round-robin method

## Transaction Affinity

When a request from mainframe to a GWSNAX(s) invoking at a transaction context, then this request is always be dispatched into same GWSNAX until the transaction is aborted or committed.

## 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 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:

**Table 3-3 crmlkon Command Options**

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.

**Table 3-3 crmlkon Command Options**

Keyword	Default	Required/Optional	Description
<linkname>	None	Required	Names the link to be started. This is the DM_SNALINKS entry in the gwsnax.cfg 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 `crmlkon` command line parameters. Refer to [“crmlkon Command”](#) for more information about the command line parameters.



**SET OBJLIB**

Indicates the name of the PDSE library where the `crmlkon` 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 `CRMLKON`. A sample member, `ENV`, is delivered with your product.

**SET SIZE**

Defines the region size for the running `crmlkon` task.

**SET ENV**

Indicates the `ENVFILE` DD name. 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 `crmlkon`.

**SET CBC**

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

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

**Listing 3-4 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 crout 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 TUXEDO MAINFRAME *
/* ADAPTER FOR SNA 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:

**Table 3-4 crmlkoff Command Options**

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.

**Table 3-4 crmlkoff Command Options**

Keyword	Default	Required/ Optional	Description
-u<keyfile>	None	Optional	Specifies a key file containing the authentication name to be used with the CRM.
<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 `crmlkoff` command line parameters. Refer to [“Sample JCL for the crmlkon Command”](#) for more information about the command line parameters.

### SET OBJLIB

Indicates the name of the PDSE library where the `crmlkoff` 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 `crmlkoff`. A sample member, `ENV`, is delivered with your product.

## SET SIZE

Defines the region size for the running `crmlkoff` task.

## SET ENV

Indicates the `ENVFILE` DD name.

## SET CEE

Specifies the high-level qualifier for the 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 3-5 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 TUXEDO MAINFRAME *
/* ADAPTER FOR SNA 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:

**Table 3-5 CRMLOGS Command Options**

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 `da1vs5:8002` and `GROUP2`, use the following command:

```
CRMLOGS GROUP2 da1vs5:8002
```

### 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.

#### SET LNKCMD

Sets the `CRMLOGS` command line parameters.

#### SET OBJLIB

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

## SET DATA1

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

## SET DATA2

Indicates the dataset containing the ENVFILE.

## SET ENVFILE

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

## SET SIZE

Defines the region size for the running CRMLOGS task.

## SET ENV

Indicates the ENVFILE DD name.

## SET CEE

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

## SET CBC

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

### Listing 3-6 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 TMA SNA 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 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 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:

**Table 3-6 crmdown Command Options**

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 3-7 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=*

//
```

---



# Command Reference Pages

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

This section discusses the following commands:

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

## CRM

Launches the Communications Resource Manager.

### Synopsis

```
CRM [ -t 0|1|2|3 ] [-p<nbr>] [-s] [-n <type>:##] [-u <keyfile>] [-V] [-P  
[interval]] [-R [interval]] [-F [trcsz]] <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 Oracle Tuxedo Mainframe Adapter for SNA 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

`-t`

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.

`-F`

Saves the spool trace copy for feeding the utility `crmmretrc`.

`trcsz`

Trace size. The value can be 10 - 2048 (M), and the default is 10.

## 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`

Encryption type, currently, the valid entries are GPE (for general purpose encryption) and SSL.

For GPE:

`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

For SSL:

`min`

Determines the minimum number of bits of cipher suite to be used for SSL encryption. They are used during the SSL negotiation between the CRM and GWSNAX. It must be less than or equal to 256. The negotiated values resolve to 40, 56, 112, 128, or 256. The strongest supported cipher suite is used finally.

`max`

Determines the maximum number of bits of cipher suite to be used for SSL encryption. They are used during the SSL negotiation between the CRM and GWSNAX. The negotiated values resolve to 40, 56, 112, 128, or 256. The strongest supported cipher suite is used finally.

## SSL Certificate Configuration Option

-S

Specifies the location of SSL configuration file. Its content is dependent on the platform on which CRM is deployed.

For CRM deployed on HP-UX:

SEC\_PRINCIPAL\_NAME, SEC\_PRINCIPAL\_LOCATION, and SEC\_PRINCIPAL\_PASSVAR are contained in the configuration file;

For CRM deployed on z/OS:

GSK\_KEYRING\_FILE, GSK\_KEYRING\_PW, GSK\_KEYRING\_LABEL are contained in the configuration file.

For more information, refer to [Using SSL Encryption](#) in Oracle Tuxedo Mainframe Adapter for SNA User's Guide.

## General Options

The following parameters apply to this command:

-s

APPC Stack API trace (default none)

addr

Socket listening address (required)

//host:port *group*

GWSNAX Group Name (required)

## Generic Resource Option

-V

Reset session affinity created by previous connection in VGR environment.

## Polling and Retrying Options

-P [*interval*]

Used for polling a link to check status of relevant partner LU; the link stops when the partner LU is disconnected.

*interval*

Polling time interval. The default value is 30 (seconds), and the minimum value is also 30.

-R [*interval*]

Used for restarting a link periodically to reconnect relevant partner LU, which has disconnected before, the link will be started when the partner LU is connected.

`interval`

Retry time interval. The default value is 30 (seconds), and the minimum value is also 30.

**Note:** `-P` and `-R` impact all CRM links.

`-P` and `-R` can be set together, but only the polling interval is valid, the retry interval is ignored. If a link is stopped by CRM utilities (for example, `CRMLKOFF`), `-R` is invalid.

## 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 Oracle Tuxedo Mainframe Adapter for SNA *Release Notes* for a list of operating systems and stacks that can communicate with the CRM.

## 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 Tuxedo Mainframe Adapter for SNA 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 Oracle Tuxedo Mainframe Adapter for SNA 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](#)

# crmretrc

Retrieve a CRM trace.

## Synopsis

```
crmretrc -n<hostname:port> [-v -c -h -u<keyfile>]
```

## Description

`crmretrc` retrieves trace from CRM, which is specified in 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 only be used on open system as CRM is usually used on MF.

-n

Names the machine and port running the CRM server.

-v

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

-c

Validates the EBCDIC to ASCII conversion.

-u

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

<keyfile>

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 retrieve the trace of CRM on mach1 at port 5000 with conversion:

```
crmretrc -c -n mach1:5000
```

## Diagnostics

`crmretrc` returns 0 upon successful completion, and returns 1 as failed with error messages.

# 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.



# Sample JCL for an MVS Platform

Job Control Language (JCL) is used on a z/OS 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 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.

## 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.

### SET OBJLIB

Indicates the name of the PDSE library where the `crmlkon` 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 `CRMLKON`. A sample member, `ENV`, is delivered with your product.

### SET SIZE

Defines the region size for the running `crmlkon` task.

### SET ENV

Indicates the `ENVFILE` DD name. 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 `crmlkon`.

### SET CBC

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

**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 TUXEDO MAINFRAME *
/* ADAPTER FOR SNA 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 `crmlkoff` command line parameters. Refer to “[crmlkoff](#)” for more information about the command line parameters.

### SET OBJLIB

Indicates the name of the PDSE library where the `crmlkoff` 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 `crmlkoff`. A sample member, `ENV`, is delivered with your product.

### SET SIZE

Defines the region size for the running `crmlkoff` task.

### SET ENV

Indicates the `ENVFILE` DD name.

### SET CEE

Specifies the high-level qualifier for the 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 TUXEDO MAINFRAME *
//* ADAPTER FOR SNA 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 `crmdown` command line parameters. Refer to [“crmdown”](#) for more information about the 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 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 [CRMLOGS](#) 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 crout 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 TMA SNA 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
//

```

---

# CRM Error Messages

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

<b>9001:ERROR</b>	<b>&lt;taskname&gt; timed out with failCode &lt;failcode&gt;</b>
<b>DESCRIPTION</b>	<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;traname&gt;, or</p> <p>IB-Conversation #nn (&lt;linkref&gt;) tx #m &lt;traname&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;traname&gt;), or</p> <p>Inbound Conversation nnnn Link (&lt;linkref&gt;) TCTXT (tctxt) Tran(&lt;traname&gt;)</p> <p>where</p> <p>nnnn is the CRM Logical Task Number and tctxt is the transaction context, where FFFFFFFF indicates non-transactional.</p>
<b>ACTION</b>	Examine <code>stderr</code> and the ULOG for additional information concerning the failure.
<b>9002:ERROR</b>	<b>Unable to create APPC Server for %s.</b>
<b>DESCRIPTION</b>	CRM was unable to instantiate the stack object due to an error.
<b>ACTION</b>	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.
<b>9003:ERROR</b>	<b>Server Failed (&lt;stackref&gt;), Code = &lt;returncode&gt;</b>
<b>DESCRIPTION</b>	CRM received a bad return code from the stack start-up.



	<b>ACTION</b>	The <returncode> is the value returned by the SNA Stack software. Check the status of the stack, the configuration of the stack, and the gateway configuration.  <b>Note:</b> Unless you started the CRM as a Tuxedo server, you must manually kill the CRM process.
<b>9004:ERROR</b>	<b>Configuration change on link &lt;linkref&gt; requires cold start</b>	
	<b>DESCRIPTION</b>	Attempting to do a warm start after changing the domain configuration.
	<b>ACTION</b>	Change start type to COLD and restart.
<b>9005:WARNING</b>	<b>Unrecovered transaction, TCTXT(%s), blob dropped. Transaction presumed forgotten.</b>	
	<b>DESCRIPTION</b>	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.
	<b>ACTION</b>	None. This message is for information only.
<b>9006:ERROR</b>	<b>Unable to start the recovery task for link &lt;linkref&gt;</b>	
	<b>DESCRIPTION</b>	An error occurred during the warm start of Tuxedo.
	<b>ACTION</b>	Cold start the Tuxedo application.
<b>9008:WARNING</b>	<b>Unknown tranid dropped, id=&lt;tctxt&gt;</b>	
	<b>DESCRIPTION</b>	Recovery was requested by Tuxedo on a transaction that was already forgotten by the CRM.
	<b>ACTION</b>	None. This message is for information only.
<b>9009:ERROR</b>	<b>No blob with recovery request. Transaction dropped, id=&lt;tctxt&gt;</b>	
	<b>DESCRIPTION</b>	Attempting to warm start after the CRMs BLOBLOG has been modified.

	<p><b>ACTION</b>                      Change start type to COLD and restart.</p>
<b>9010:ERROR</b>	<b>&lt;taskname&gt; failed with failCode &lt;failcode&gt;</b>
<p><b>DESCRIPTION</b></p>	<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>
<p><b>ACTION</b></p>	<p>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 <code>BLOBLOG</code> and <code>RSTRTLOG</code>. If no apparent error is found, contact Oracle Customer Support.</p>

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

<b>9016:ERROR</b>	<b>Link refers to undefined APPC stack (&lt;stackref&gt;!)</b>
<b>DESCRIPTION</b>	The <code>stackref</code> in the link configuration is incorrect.
<b>ACTION</b>	Correct the <code>stackref</code> that is in error, run <code>dmloadcf</code> , and restart.
<b>9017:ERROR</b>	<b>INTERNAL ERROR: link registration failed</b>
<b>DESCRIPTION</b>	Internal error registering the link. The stack failed.
<b>ACTION</b>	Contact Oracle Customer Support.
<b>9018:ERROR</b>	<b>Invalid Transition &lt;additional information&gt;</b>
<b>DESCRIPTION</b>	An internal state table failure has occurred. The <additional information> will be one of the following: <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>
<b>ACTION</b>	Contact Oracle Customer Support.
<b>9019:ERROR</b>	<b>Unknown Service Correlator = &lt;correlator&gt;, message dropped</b>
<b>DESCRIPTION</b>	Internal error assigning service correlator values. Message context lost.
<b>ACTION</b>	Contact Oracle Customer Support.
<b>9020:ERROR</b>	<b>Duplicate Service Correlator = &lt;correlator&gt;</b>
<b>DESCRIPTION</b>	Internal error assigning service correlator values.
<b>ACTION</b>	Contact Oracle Customer Support.

<b>9021:ERROR</b>	<b>Invalid Remote Link Name &lt;linkref&gt;</b>
	<b>DESCRIPTION</b> The remote link name in a request does not match any defined link name.
	<b>ACTION</b> Correct the gateway configuration and restart.
<b>9022:ERROR</b>	<b>Invalid transaction context = &lt;tctx&gt;</b>
	<b>DESCRIPTION</b> Internal error assigning transaction context values. Bad transaction.
	<b>ACTION</b> Contact Oracle Customer Support.
<b>9023:ERROR</b>	<b>Unknown Service Correlator = &lt;correlator&gt;, message dropped</b>
	<b>DESCRIPTION</b> Internal error assigning service correlator values. Message context lost.
	<b>ACTION</b> Contact Oracle Customer Support.
<b>9024:ERROR</b>	<b>Invalid initial syncpoint received from subordinate, TCTX(%s)</b>
	<b>DESCRIPTION</b> Syncpoint processing protocol violation. Subordinate member of conversation attempted to initiate a syncpoint. XA does not support syncpoints from subordinate members.
	<b>ACTION</b> Contact Oracle Customer Support.
<b>9025:ERROR</b>	<b>Invalid Input Message Discarded</b>
	<b>DESCRIPTION</b> Internal error, bad message sent between GWSNAX and CRM. Possibly incompatible Tuxedo Mainframe Adapter for SNA Gateway and CRM.
	<b>ACTION</b> Contact Oracle Customer Support.
<b>9026:ERROR</b>	<b>CNOS Notification Received for unknown partner &lt;partnerLU&gt;</b>
	<b>DESCRIPTION</b> Multiple instances of the CRM may be using the same local LU.
	<b>ACTION</b> Ensure that multiple Tuxedo configurations do not use the same local LU.

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

<b>9033:ERROR</b>	<b>Requested Synclevel not supported by link &lt;linkref&gt; for context &lt;correlator&gt; (synclevel &lt;level&gt;)</b>
<b>DESCRIPTION</b>	Attempted to issue a request at sync level <level> on a link that does not support that level.
<b>ACTION</b>	Correct application or gateway configuration.
<b>9034:ERROR</b>	<b>Service Request at SyncLevel=2 Rejected on PENDING link &lt;linkref&gt; for context &lt;correlator&gt;</b>
<b>DESCRIPTION</b>	An attempt to start a new sync level 2 request has been received and the Link is currently processing recovery information.
<b>ACTION</b>	Wait until recovery is complete to request sync level 2 services.
<b>9035:ERROR</b>	<b>Inbound Request Transform Failed (&lt;status&gt;) for context &lt;correlator&gt;</b>
<b>DESCRIPTION</b>	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.
<b>ACTION</b>	Check gateway configuration for incorrect specification of local service API entry.
<b>9036:ERROR</b>	<b>Inbound Response Transform Failed (&lt;status&gt;) for context &lt;correlator&gt;</b>
<b>DESCRIPTION</b>	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.
<b>ACTION</b>	Check gateway configuration for incorrect specification of local service API entry.

<b>9037:ERROR</b>	<b>Outbound Request Transform Failed (&lt;status&gt;) for context &lt;correlator&gt;</b>
<b>DESCRIPTION</b>	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.
<b>ACTION</b>	Check gateway configuration for incorrect specification of local service API entry.
<b>9038:ERROR</b>	<b>Outbound Response Transform Failed (&lt;status&gt;) for context &lt;correlator&gt;</b>
<b>DESCRIPTION</b>	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.
<b>ACTION</b>	Check gateway configuration for incorrect specification of local service API entry.
<b>9039:ERROR</b>	<b>Conversation terminated without confirm for context &lt;correlator&gt;</b>
<b>DESCRIPTION</b>	Sync level 2 conversation was terminated with out confirm.
<b>ACTION</b>	Check application program and correct.
<b>9040:ERROR</b>	<b>Inbound Confirm not supported</b>
<b>DESCRIPTION</b>	Host application is requesting an inbound confirm. This is not supported.
<b>ACTION</b>	Check host application program and correct.
<b>9041:ERROR</b>	<b>Inbound Confirm for multi-ISRT not supported</b>
<b>DESCRIPTION</b>	Host IMS application is requesting an inbound confirm and using multiple ISRT commands. This is not supported.



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

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

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

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

<b>9069 ERROR</b>	<b>CRM encryption setup failed</b>
<b>DESCRIPTION</b>	An error has occurred while establishing link-level encryption with the CRM.
<b>ACTION</b>	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
<b>9072 ERROR</b>	<b>Attempted access by unauthorized CRM client</b>
<b>DESCRIPTION</b>	A client has attempted to access the CRM without the proper authentication or encryption setup.
<b>ACTION</b>	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.
<b>9073 ERROR</b>	<b>CRM authentication setup failed</b>
<b>DESCRIPTION</b>	An error has occurred while authenticating a CRM client.
<b>ACTION</b>	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.
<b>9074 ERROR</b>	<b>CRM Logical Unit %s is probably inactive or unconfigured</b>
<b>DESCRIPTION</b>	An error has occurred while connecting to the APPC stack.
<b>ACTION</b>	Verify that the CRM Logical Unit name is correct and is defined and active in both the stack and VTAM configuration.
<b>9075 WARNING</b>	<b>WARN: Synclevl on link %s</b>

	<b>DESCRIPTION</b>	The configured Synclevel could not be negotiated.
	<b>ACTION</b>	Verify that the Remote LU is configured for the correct synclevel.
<b>9076 ERROR</b>	<b>CRM Logical Unit %s is probably the wrong type</b>	
	<b>DESCRIPTION</b>	An error has occurred while connecting to the APPC stack.
	<b>ACTION</b>	Verify that the CRM Logical Unit is correctly configured.
<b>9077 ERROR</b>	<b>CRM Logical Unit %s is already in use</b>	
	<b>DESCRIPTION</b>	An error has occurred while connecting to the APPC stack.
	<b>ACTION</b>	Verify that the CRM Logical Unit is not in use by another application.
<b>9079 ERROR</b>	<b>CRM client message rejected -- incompatible software/protocol version (CRM API RC=%s)</b>	
	<b>DESCRIPTION</b>	A CRM client has attempted to communicate with the CRM, but the client is at a software version level that is incompatible with and unsupported by the CRM.
	<b>ACTION</b>	The CRM rejects the message and terminates the connection. Check the version levels of the CRM and the client to ensure compatibility.
<b>9080 ERROR</b>	<b>Gateway Signon request rejected -- CRM disconnect in progress</b>	
	<b>DESCRIPTION</b>	The CRM is in the process of stopping all links and de-configuring, and is not able to accept a new Signon request at this time.
	<b>ACTION</b>	The Signon request is rejected. When the CRM has completed disconnect processing and returned to the reset state, retry the Signon request.
<b>9081 ERROR</b>	<b>Failure occurred during Rollback of %s, remote resources may not be backed out</b>	

	<p><b>DESCRIPTION</b>      An Error occurred while performing BACKEDOUT processing. Remote resources may not be backed out.</p>
	<p><b>ACTION</b>              Check mainframe application transaction logs to determine the cause and take the appropriate action.</p>
<p><b>9082 INFO</b></p>	<p><b>Unable to determine if the transaction was committed on the partner side, possible heuristic situation</b></p>
	<p><b>DESCRIPTION</b>      Failed to determine the state of the global transaction due to a partial resynchronization.</p>
	<p><b>ACTION</b>              Examine the CRM traces for more information concerning the reason for the failure. Check the CICS logs to see the transaction status.</p>





# 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 Oracle 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.

## **B**

(No terms begin with the letter “B.”)

## **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 Oracle 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 Oracle 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.

## **G**

(No terms begin with the letter “G.”)

## **H**

(No terms begin with the letter “H.”)

## **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.

## **J**

(No terms begin with the letter “J.”)

## **K**

(No terms begin with the letter “K.”)

## **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 Oracle Tuxedo Mainframe Adapter for SNA 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**

Oracle Tuxedo Mainframe Adapter for SNA feature which enables Oracle Tuxedo clients to invoke host CICS/ESA programs and, conversely, CICS/ESA client programs to invoke Oracle 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.

## N

(No terms begin with the letter “N.”)

## 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.

## Q

(No terms begin with the letter “Q.”)

## 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 Oracle Tuxedo System/*T application* is bounded by the environment described in a single `TUXCONFIG` file. In `/Domain`, a Oracle 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.

## **U**

(No terms begin with the letter “U.”)

## **V**

### **Virtual Telecommunications Access Method (VTAM)**

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

## **W**

(No terms begin with the letter “W.”)

## **X**

(No terms begin with the letter “X.”)

## **Y**

(No terms begin with the letter “Y.”)

## **Z**

(No terms begin with the letter “Z.”)

