

Oracle® Tuxedo

Mainframe Adapter for SNA User's Guide



Release 22c

F85267-02

January 2024

The Oracle logo, consisting of a solid red square with the word "ORACLE" in white, uppercase, sans-serif font centered within it.

ORACLE®

Oracle Tuxedo Mainframe Adapter for SNA User's Guide, Release 22c

F85267-02

Copyright © 1996, 2024, Oracle and/or its affiliates.

Primary Author: Preeti Gandhe

Contributing Authors: Tulika Das

Contributors: Maggie Li

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, software documentation, data (as defined in the Federal Acquisition Regulation), or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. 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®, Java, MySQL and NetSuite are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about 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 unless otherwise set forth in an applicable agreement between you and Oracle. 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, except as set forth in an applicable agreement between you and Oracle.

Contents

Preface

Documentation Accessibility	xiv
-----------------------------	-----

1 Oracle Tuxedo Mainframe Adapter for SNA Overview

2 The Oracle Tuxedo Mainframe Adapter for SNA Architecture

2.1 The Oracle Tuxedo Mainframe Adapter for SNA Gateway	2-2
2.2 Communications Resource Manager	2-2

3 Configuring and Starting the System

3.1 Preparing for Configuration	3-1
3.1.1 Determine Your System Architecture	3-1
3.1.1.1 Oracle Tuxedo Mainframe Adapter for SNA Components	3-1
3.1.1.2 System Configuration	3-2
3.1.2 Configure the Local Host	3-4
3.1.3 Configure the Remote Host	3-4
3.2 Configuring the Oracle Tuxedo Mainframe Adapter for SNA Gateway	3-4
3.2.1 Overview	3-4
3.2.1.1 Step 1: Edit the DMATYPE File	3-5
3.2.1.2 Step 2: Edit the UBBCONFIG File	3-5
3.2.1.3 Step 3: Edit the DMCONFIG File (and Edit CRM Configuration File to Have Separate CRM Configurations on Mainframe)	3-6
3.2.2 Configurations for Customized Call ID Support	3-13
3.2.2.1 Set Environment Variable CALLID_FIELDS_CFG	3-14
3.2.2.2 Set Configuration File CALLFLDS	3-14
3.2.2.3 Sample	3-15
3.3 Using the Oracle Tuxedo Metadata Repository	3-16
3.3.1 Tuxedo Metadata Respository Service-Level Keywords	3-17
3.3.2 Mapping Between DMCONFIG Elements and Tuxedo Metadata Repository Keywords	3-17
3.3.3 GWSNAX Server Configuration	3-18

3.4	Starting the System	3-18
3.4.1	Step 1: Start the CRM	3-18
3.4.2	Step 2: Start the ATMI Servers	3-18

4 Verifying the Software

4.1	Building Verification Tests	4-1
4.1.1	Building ATMI Platform Executables	4-2
4.1.1.1	Modify the UBBCONFIG File	4-2
4.1.1.2	Execute the tmlodcf Command	4-3
4.1.1.3	Modify the DMCONFIG File	4-3
4.1.1.4	Execute the dmlodcf Command	4-4
4.1.1.5	Modify the Environment Files	4-4
4.1.1.6	Build the Server	4-6
4.1.1.7	Build the Client	4-6
4.1.2	Building CICS/ESA Executables	4-6
4.1.2.1	Choose the Source Code Language	4-7
4.1.2.2	Transfer the Source Code to the Host	4-7
4.1.2.3	Translate CICS/ESA Verbs	4-8
4.1.2.4	Compile the Translated Source File	4-9
4.1.2.5	Create the Executable Object	4-10
4.1.2.6	Configure the CICS/ESA Application	4-12
4.1.2.7	View Connection and Session Status	4-16
4.2	Running the Sample Application	4-16
4.2.1	Running the Application from an ATMI Client	4-17
4.2.2	Running the Application from a CICS/ESA Client	4-18
4.2.2.1	CICS/ESA Client with CPI-C	4-18
4.2.2.2	CICS/ESA Client with DPL	4-18
4.2.3	Running the Application from a CICS/VSE Client	4-19
4.2.3.1	CICS/VSE Sample Applications	4-19
4.2.3.2	Other Considerations	4-19

5 Providing Security

5.1	Understanding Oracle Tuxedo Mainframe Adapter for SNA Security	5-1
5.2	Mapping User IDs	5-1
5.2.1	ATMI-to-Host User ID Mapping	5-3
5.2.2	Direct User ID Mapping	5-3
5.2.3	Configuring User ID Mapping	5-4
5.2.3.1	Determining Security Parameters	5-4
5.2.3.2	Setting DMCONFIG File Security Parameters	5-8

5.2.3.3	Setting UBBCONFIG File Security Parameters	5-10
5.2.3.4	Bypassing User ID Mapping	5-11
5.2.4	Using dmadm Commands to Administer User ID Mapping	5-11
5.2.4.1	Adding a User ID and Password	5-12
5.2.4.2	Mapping a User ID	5-12
5.2.4.3	Removing User ID Mapping	5-13
5.2.4.4	Deleting a User ID and Password	5-13
5.2.4.5	Modifying a Password	5-14
5.2.5	Setting Security Scenario	5-14
5.2.5.1	Configuring Security in the ATMI Domain	5-14
5.2.5.2	Configuring Security in the Local Domain	5-16
5.2.5.3	Configuring Security in the Remote Domain	5-16
5.2.5.4	Setting the Security Level to IDENTIFY	5-17
5.3	Using Link-Level Encryption	5-17
5.3.1	Illustration of Encryption Process	5-17
5.3.2	Configuring the Oracle Tuxedo Mainframe Adapter for SNA Gateway and CRM	5-18
5.4	Using TLS Encryption	5-18
5.4.1	Configuring the Oracle Tuxedo Mainframe Adapter for SNA Gateway and CRM	5-19
5.4.2	Security Enforcement	5-19
5.4.3	Samples of TLS Encryption Configuration File for CRM	5-20
5.5	Using TCP/IP Link Authentication	5-20
5.5.1	Illustration of Authentication Process	5-21
5.5.2	Configuring the Oracle Tuxedo Mainframe Adapter for SNA Gateway and CRM for Authentication	5-21

6 Setting Up Data Translations

6.1	Data Conversion	6-1
6.1.1	Conversion of ATMI Typed Buffers to Records	6-2
6.1.1.1	Data Conversion for STRING Typed Buffer	6-2
6.1.1.2	Data Conversion for XML Typed Buffer	6-2
6.1.1.3	Data Conversion for X_OCTET/CARRAY Typed Buffers	6-3
6.1.1.4	Data Conversion for VIEW/VIEW32/X_C_TYPE/X_COMMON Typed Buffers	6-3
6.1.1.5	Data Conversion for FML/FML32 Typed Buffers	6-3
6.1.1.6	Data Conversion for RECORD Typed Buffers	6-3
6.1.2	Conversion of Records to ATMI Typed Buffers	6-3
6.1.2.1	Data Conversion for STRING Typed Buffer	6-4
6.1.2.2	Data Conversion for XML Typed Buffer	6-4
6.1.2.3	Data Conversion for X_OCTET/CARRAY Typed Buffers	6-4
6.1.2.4	Data Conversion for VIEW/VIEW32/X_C_TYPE/X_COMMON Typed Buffers	6-4

6.1.2.5	Data Conversion for FML/FML32 Typed Buffers	6-5
6.1.2.6	Data Conversion for RECORD Typed Buffers	6-5
6.1.3	Data Conversion For DPL Services	6-5
6.1.3.1	DPL Requests Originating From an ATMI Application	6-6
6.1.3.2	DPL Requests Originating From a CICS DPL	6-6
6.2	Translation Rules for VIEW Data Types	6-6
6.2.1	String Considerations	6-8
6.2.2	Converting Numeric Data	6-8
6.3	Translation Rules for RECORD Data Types	6-8
6.4	Translation Rules for Strings	6-8
6.4.1	Setting the Option to Perform String Transformation	6-10
6.4.2	Code Page Translation Tables	6-11
6.4.2.1	Specifying a Translation Table	6-12
6.4.2.2	How the Translation Tables Work	6-13
6.5	Multi-byte Character Set (MBCS) Support	6-14
6.6	ART for CICS Integration Consideration	6-14

7 APPC/IMS Programming Considerations

7.1	APPC/IMS Overview	7-1
7.1.1	Implicit API	7-1
7.1.2	Explicit API	7-2
7.2	APPC/IMS Programming	7-2
7.2.1	Non-Transactional Application Programming	7-2
7.2.2	Transactional Application Programming	7-4
7.2.3	Sample Transaction Programs	7-4

8 Channel/Container Support in DPL

8.1	Channel / Container Overview	8-1
8.2	ATMI Server Programming	8-1
8.3	Configurations	8-2
8.3.1	VIEW32 Definitions	8-2
8.3.2	DMCONFIG	8-2
8.3.3	Environment Variable	8-3

9 Inbound Transaction Routing

9.1	Overview	9-1
9.2	Configuration	9-1
9.2.1	MF CICS Configuration	9-1
9.2.2	ARTCICS and TMA SNA Configuration	9-2

9.3	Example(s)	9-2
9.3.1	Configuration	9-3

10 TMA/SNA High Availability

10.1	Outbound Call (from Oracle Tuxedo to CICS)	10-1
10.1.1	Single Connection and Multiple Links	10-1
10.1.1.1	Configuring UBBCONFIG	10-2
10.1.1.2	Configuring DMCONFIG	10-3
10.1.1.3	Configuring VTAM List	10-5
10.1.1.4	Configuring Connections	10-6
10.1.1.5	Programming for DPLS	10-6
10.1.2	Multiple Connections and Multiple Links	10-7
10.2	Inbound Call (from CICS to Oracle Tuxedo)	10-8
10.2.1	Single Connection and Multiple Links	10-9
10.2.1.1	Configuring UBBCONFIG	10-10
10.2.1.2	Configuring DMCONFIG	10-11
10.2.1.3	Configuring VTAM List	10-12
10.2.1.4	Configuring Connections	10-13
10.2.1.5	Programming for DPLC	10-14
10.2.1.6	Configuring CICS System Initialization Table	10-15
10.2.2	Multiple Connections and Multiple Links	10-15

11 ART CICS Calls Transaction in Z/CICS

11.1	Step 1: Configuring ART CICS on Open System	11-1
11.2	Step 2: Defining VTAM Resources	11-1
11.3	Step 3: Configuring and Deploying TMA SNA	11-3
11.4	Step 4: Start CRM on Mainframe	11-6

12 ART IMS Calls Transaction in Z/IMS

12.1	Step 1: Configuring Sub-transactions on Z/OS	12-1
12.2	Step 2: Configuring and Deploying TMA SNA for IMS	12-2
12.3	Step 3: Generating Buffer Converter Program for Sub-transactions	12-4

A Administrative Command Reference Pages

A.1	addumap	A-1
A.1.1	Synopsis	A-2
A.1.2	Description	A-2
A.1.3	Portability	A-2

A.1.4	Diagnostics	A-2
A.1.5	Example	A-3
A.2	addusr	A-3
A.2.1	Synopsis	A-3
A.2.2	Description	A-3
A.2.3	Portability	A-4
A.2.4	Diagnostics	A-4
A.2.5	Examples	A-4
A.3	delumap	A-4
A.3.1	Synopsis	A-5
A.3.2	Description	A-5
A.3.3	Portability	A-5
A.3.4	Diagnostics	A-5
A.3.5	Example	A-6
A.4	delusr	A-6
A.4.1	Synopsis	A-6
A.4.2	Description	A-6
A.4.3	Portability	A-7
A.4.4	Diagnostics	A-7
A.4.5	Examples	A-7
A.5	DMADM	A-7
A.5.1	Synopsis	A-7
A.5.2	Description	A-8
A.5.3	Portability	A-8
A.5.4	Examples	A-8
A.6	dmdadmin	A-8
A.6.1	Synopsis	A-9
A.6.2	Description	A-9
A.6.3	Administration Mode Commands	A-10
A.6.4	Configuration Mode Commands	A-13
A.6.5	Configuration Input Format	A-16
A.6.6	Configuration Limitations	A-16
A.6.7	Restrictions for Configuration Field Identifiers/Updates	A-16
A.6.8	Configuring the DM_LOCAL_DOMAINS Section	A-17
A.6.9	Configuring the DM_REMOTE_DOMAINS Section	A-17
A.6.10	Configuring the DM_TDOMAIN Section	A-18
A.6.11	Configuring the DM_OSITP Section	A-18
A.6.12	Configuring the DM_LOCAL_SERVICES Section	A-18
A.6.13	Configuring the DM_REMOTE_SERVICES Section	A-19
A.6.14	Configuring the DM_ROUTING Section	A-19
A.6.15	Configuring the DM_ACCESS_CONTROL Section	A-20

A.6.16	Configuring the DM_PASSWORDS Section	A-20
A.6.17	Diagnostics in Configuration Mode	A-20
A.6.18	Configuration Example	A-22
A.6.19	Security	A-23
A.6.20	Environment Variables	A-23
A.6.21	General Diagnostics	A-24
A.6.22	Interoperability	A-24
A.6.23	Portability	A-24
A.7	dmconfig	A-24
A.7.1	Description	A-25
A.7.2	Definitions	A-25
A.7.3	Configuration File Format	A-26
A.7.4	The DM_LOCAL_DOMAINS Section	A-27
A.7.5	The DM_REMOTE_DOMAINS Section	A-30
A.7.6	The DM_TDOMAIN Section	A-31
A.7.7	The DM_OSITP Section	A-31
A.7.8	The DM_SNACRM Section	A-33
A.7.9	The DM_SNASTACKS Section	A-34
A.7.10	The DM_SNALINKS Section	A-35
A.7.11	The DM_ACCESS_CONTROL Section	A-37
A.7.12	The DM_LOCAL_SERVICES Section	A-37
A.7.13	The DM_REMOTE_SERVICES Section	A-38
A.7.14	The DM_ROUTING Section	A-41
A.7.15	Files	A-42
A.7.16	Example 1	A-42
A.7.17	Example 2	A-44
A.7.18	Example 3	A-45
A.8	dmloadcf	A-47
A.8.1	Synopsis	A-47
A.8.2	Description	A-47
A.8.3	Portability	A-49
A.8.4	Environment Variables	A-49
A.8.5	Examples	A-49
A.8.6	Diagnostics	A-49
A.9	dmunloadcf	A-50
A.9.1	Synopsis	A-50
A.9.2	Description	A-50
A.9.3	Portability	A-50
A.9.4	Examples	A-50
A.9.5	Diagnostics	A-50
A.10	GWADM	A-51

A.10.1	Synopsis	A-51
A.10.2	Description	A-51
A.10.3	Portability	A-52
A.10.4	Interoperability	A-52
A.10.5	Examples	A-52
A.11	GWSNAX	A-53
A.11.1	Synopsis	A-53
A.11.2	Description	A-53
A.11.3	Portability	A-55
A.11.4	Interoperability	A-55
A.11.5	Examples	A-55
A.12	modusr	A-56
A.12.1	Synopsis	A-56
A.12.2	Description	A-56
A.12.3	Portability	A-57
A.12.4	Diagnostics	A-57
A.12.5	Examples	A-57
A.13	tmadmin	A-57
A.13.1	tmadmin(1)	A-57
A.13.1.1	Name	A-58
A.13.1.2	Synopsis	A-58
A.13.1.3	Description	A-58
A.13.2	tmadmin Command Related to TMA	A-59
A.13.2.1	Examples	A-60

B Error Messages

C Code Page Translation Tables

C.1	Modifying a Code Page Translation Table	C-1
C.2	Default Tuxedo Code Page Translation Table	C-2
C.3	United States (00819x00037) Code Page Translation Table	C-3
C.4	Germany (00819x00273) Code Page Translation Table	C-4
C.5	Finland/Sweden (00819x00278) Code Page Translation Table	C-5
C.6	Spain (00819x00284) Code Page Translation Table	C-7
C.7	Great Britain (00819x00285) Code Page Translation Table	C-8
C.8	France (00819x00297) Code Page Translation Table	C-9
C.9	Belgium (00819x00500) Code Page Translation Table	C-10
C.10	Portugal (00819x00860) Code Page Translation Table	C-11
C.11	Italy (00819x00280) Code Page Translation Table	C-12

C.12	Latin-1 – (00819x01047) Code Page Translation Table	C-13
C.13	Latin-2 – (00912x00870) Code Page Translation Table	C-14

Index

List of Figures

2-1	Tuxedo Mainframe Adapter for SNA Architecture	2-1
3-1	Local Oracle Tuxedo Mainframe Adapter for SNA Configuration on Unix/Linux Platform	3-3
3-2	Distributed CRM on z/OS Platform	3-3
3-3	Distributed CRM on UNIX/Linux Platform	3-4
5-1	Oracle Tuxedo Mainframe Adapter for SNA Security Elements	5-2
5-2	Typical ATMI-to-host User ID Mapping	5-3
5-3	Direct User ID Mapping	5-4
5-4	Encrypted Links	5-17
5-5	Authenticated Links	5-21
6-1	Oracle Tuxedo Mainframe Adapter for SNA Code Page Translation	6-12
9-1	Inbound Transaction Routing Support Configuration	9-1
9-2	Example 1	9-2
10-1	Outbound Call (Single Connection and Multiple Links)	10-1
10-2	Multiple Connections and Multiple Links	10-7
10-3	Inbound Call (from CICS to Oracle Tuxedo)	10-9
10-4	Multiple Connections and Multiple Links	10-15

List of Tables

3-1	CALLFLDS Parameters	3-15
4-1	Source Code Filenames	4-7
5-1	Security Settings for Inbound Requests from Host Systems	5-5
5-2	Security Settings for Outbound Requests from Local Domain	5-7
5-3	CRM Processes Supporting Authentication	5-20
6-1	DPL Request LENGTH Calculation	6-6
6-2	VIEW Data Translation Rules	6-7
6-3	Data Translation Rules between C and IBM Z Data Types	6-7
6-4	C to COBOL String Transformation	6-10
6-5	Oracle Tuxedo Mainframe Adapter for SNA Code Page Translation Tables	6-13
8-1	CICS Container and FML32 APIs Mapping	8-1
8-2	CICS Command Options and FML32 Fields Mapping	8-2
10-1	Configurations for Single Connection and Multiple Links (Outbound)	10-2
10-2	Configurations for Multiple Connection and Multiple Links (Outbound)	10-8
10-3	Configurations for Single Connection and Multiple Links (Inbound)	10-9
A-1	DM_LOCAL_DOMAINS SECTION	A-17
A-2	DM_REMOTE_DOMAINS SECTION	A-17
A-3	DM_TDOMAIN SECTION	A-18
A-4	DM_OSITP SECTION	A-18
A-5	DM_LOCAL_SERVICES SECTION	A-18
A-6	DM_REMOTE_SERVICES SECTION	A-19
A-7	DM_ROUTING SECTION	A-19
A-8	DM_PASSWORDS SECTION	A-20
A-9	DM_PASSWORDS SECTION	A-20
A-10	A-10 C to COBOL String Transformation	A-55

Preface

In today's rapidly changing environment, businesses must continue to address changes in their business needs through adjustments in their corresponding computer infrastructure. Packaged applications successfully automate many internal operations such as financial, manufacturing, and human resources tasks. However, these applications' data formats and interface protocols may be proprietary and much of the applications' functionality may not be accessible from any exposed Application Programming Interface (API). The consequence is isolation of these systems, loss of flexibility, and the inability to change at a later time.

When business systems need to share information and capabilities to operate efficiently or to expand to the web, integration problems surface. Businesses need a greater exchange of information with systems that communicate at both a database and a process level within the organization as well as with customers' and suppliers' systems. Businesses need to develop systems that are open, robust, and flexible to change while retaining the systems they have already purchased, developed, or inherited.

- [Documentation Accessibility](#)

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

1

Oracle Tuxedo Mainframe Adapter for SNA Overview

Oracle Tuxedo Mainframe Adapter for SNA allows Application-to-Transaction Monitor Interface (ATMI) applications to communicate with the mainframe. The Oracle Tuxedo Mainframe Adapter products provides domain-compliant gateways that permit administration of the remote *Transaction Processing (TP)* system as a foreign domain.

The Tuxedo Mainframe Adapter domain architecture extends the scope of ATMI platforms to provide coordinated transaction processing across an enterprise's geographic or organizational boundaries. Within each domain, the administrator determines which local services are available to other specific domains, thus enabling client applications to request those services.

The domain gateway architecture is designed for the ATMI application administrator, who makes services in other domains available to application programmers. The existence of applications within distinct domains is, however, totally transparent to the application programmers. They can use ATMI programming paradigms to request services offered in other domains as if they were services offered within the local application.

The ATMI application administrator enables remote domains to access a subset of *Local Services*. This subset is called a *Local Domain*. The local domain helps the administrator provide secure “views” of the application.

The Oracle Tuxedo Mainframe Adapter for SNA Gateway communicates between independent Logical Units (LUs) using LU6.2 sessions and conversations. It adds multi-domain connectivity, bidirectional request/response, and conversations between ATMI platforms and System Network Architecture (SNA)-based applications. It provides support for DPL and DTP transactions in a CICSplex environment. It provides access to Application Program-to-Program Communication (APPC) applications based on SNA as well as inter-system communication with Customer Information Control System/Enterprise System Architecture (CICS/ESA). When accessing CICS/ESA systems, the Oracle Tuxedo Mainframe Adapter for SNA acts as a CICS/ESA region, capable of supporting the following sync-level 2 functions:

- Bidirectional request/response and conversational service requests
- Peer support for CICS/ESA and support for multiple remote MVS regions
- Event monitoring and reporting
- Both static and dynamic configuration support where allowed by SNA
- Automatic data conversion between UNIX and host formats
- Security through Access Control Lists on the local domain
- User and password security for CICS/ESA and IMS communications
- Security access through plug-in for third-party security software

2

The Oracle Tuxedo Mainframe Adapter for SNA Architecture

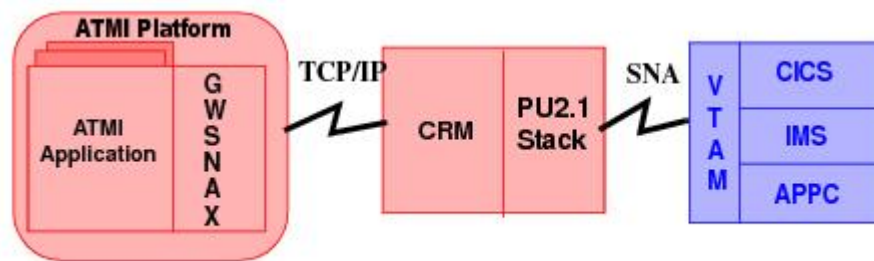
Oracle Tuxedo Mainframe Adapter for SNA is composed of two major components that can be configured to provide SNA solutions, the Gateway and the Communications Resource Manager (CRM). These components provide a bidirectional link enabling ATMI platforms such as Oracle Tuxedo to interact with mainframe applications as either a client or server, using a CICS/ESA and IMS implicit LU6.2, or any IBM-supported *Application Program-to-Program Communication (APPC)* or CICS/ESA interface. The following figure illustrates the Oracle Tuxedo Mainframe Adapter for SNA architecture.

The Gateway (GWSNAX) runs on an ATMI platform environment and the Communications Resource Manager (CRM) can run in the native UNIX environment or be distributed to the mainframe as a Virtual Telecommunications Access method (VTAM) application. The CRM may also be distributed to another UNIX system, separate from the ATMI platform.

Generally, the Oracle Tuxedo Mainframe Adapter for SNA domain is like other ATMI domain gateways. It uses the `DMADM` and `GWADM` servers for administration. Within the Oracle Tuxedo Mainframe Adapter for SNA system, additional servers and processes support peer CICS/ESA and IMS connectivity and sync level 2.

Note:
An ATMI platform is required, and a third-party SNA stack is required if the CRM is not installed on a mainframe. These products are sold separately.

Figure 2-1 Tuxedo Mainframe Adapter for SNA Architecture



- [The Oracle Tuxedo Mainframe Adapter for SNA Gateway](#)
- [Communications Resource Manager](#)

2.1 The Oracle Tuxedo Mainframe Adapter for SNA Gateway

The Oracle Tuxedo Mainframe Adapter for SNA Gateway is an ATMI domain gateway that communicates with the CRM. The Gateway processes ATMI-to-mainframe requests and responses in conjunction with the CRM. Requests coming from the mainframe are mapped to ATMI services, while requests originating in ATMI are mapped to mainframe programs that can be executed using a CICS Distributed Program Link (DPL) or Distributed Transaction Processing (DTP) application, or started from an Information Management System (IMS) queue.

2.2 Communications Resource Manager

The CRM runs as a separate native process. It enables APPC conversations and DPL protocols to flow into and out of the ATMI environment. The CRM obtains its configuration from the Gateway. If the Gateway is running on a platform other than the one on which the CRM is running, the CRM should be started before the ATMI platform is started so it will be monitoring the address specified in the Gateway configuration.

A properly configured SNA protocol stack is required for the CRM to communicate with a mainframe, unless the CRM is running independently of the ATMI environment (distributed mode).

3

Configuring and Starting the System

To use Oracle Tuxedo Mainframe Adapter for SNA, proper configuration based on your system's architecture is required.



Note:

All references to ATMI files, functions, and documentation apply to Tuxedo files, functions, and documentation.

This chapter contains the following topics:

- [Preparing for Configuration](#)
- [Configuring the Oracle Tuxedo Mainframe Adapter for SNA Gateway](#)
- [Using the Oracle Tuxedo Metadata Repository](#)
- [Starting the System](#)

3.1 Preparing for Configuration

Before you can properly configure your Oracle Tuxedo Mainframe Adapter for SNA Gateway and the CRM to communicate with CICS and IMS applications, you must complete the following prerequisites:

- [Determine Your System Architecture](#)
- [Configure the Local Host](#)
- [Configure the Remote Host](#)
- [Determine Your System Architecture](#)
- [Configure the Local Host](#)
- [Configure the Remote Host](#)

3.1.1 Determine Your System Architecture

To determine your system's architecture, you must determine the location of the Oracle Tuxedo Mainframe Adapter for SNA components within that architecture.

- [Oracle Tuxedo Mainframe Adapter for SNA Components](#)
- [System Configuration](#)

3.1.1.1 Oracle Tuxedo Mainframe Adapter for SNA Components

The following basic components of the Oracle Tuxedo Mainframe Adapter for SNA system are factors in configuring your system:

- **Oracle Tuxedo Mainframe Adapter Gateway (GWSNAX)**
The Oracle Tuxedo Mainframe Adapter Gateway is the transactional SNA gateway. It is implemented as an ATMI domain gateway and uses the ATMI environments. The gateway communicates over a Transmission Control Protocol/Internet Protocol (TCP/IP) connection with the CRM.
- **CRM**
The Communications Resource Manager (CRM) communicates with the SNA network using an SNA *stack*. It communicates with Oracle Tuxedo Mainframe Adapter clients through the Gateway.
- **SNA stack**
The stack is vendor-supplied software that provides connectivity to an SNA network.

3.1.1.2 System Configuration

Your system architecture reflects one of the following basic Oracle Tuxedo Mainframe Adapter for SNA configurations:

- [Local Configuration](#)
- [Distributed Configuration—CRM on z/OS Host](#)
- [Distributed Configuration—CRM on Unix/Linux](#)

3.1.1.2.1 Local Configuration

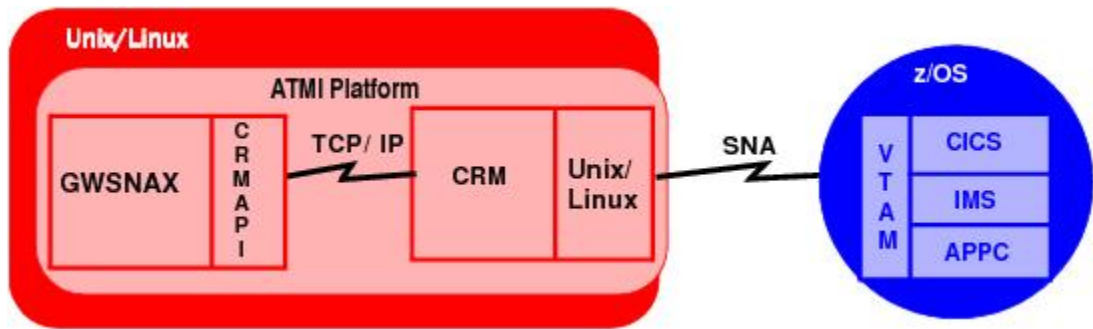
Local configuration combines the Tuxedo ATMI platform, the Oracle Tuxedo Mainframe Adapter Gateway, CRM, and SNA stack (PU2.1 server) on the same UNIX platform, as shown in the following figure. Local configuration features the widely used TCP/IP connectivity between the Oracle Tuxedo Mainframe Adapter for SNA Gateway and CRM, giving a high-performance communications interface. On the mainframe side, the CRM uses a stack to communicate over a System Network Architecture (SNA) interface with the host system. This configuration allows you to:

- Connect to an existing SNA network.
- Consolidate the ATMI platform and Oracle Tuxedo Mainframe Adapter for SNA systems on the same platform.
- Consolidate the ATMI platform and Oracle Tuxedo Mainframe Adapter for SNA systems on the same platform.

 **Note:**

A one-to-one relationship exists between the Oracle Tuxedo Mainframe Adapter Gateway and CRM. The Oracle Tuxedo Mainframe Adapter Gateway cannot be configured to handle multiple CRM processes.

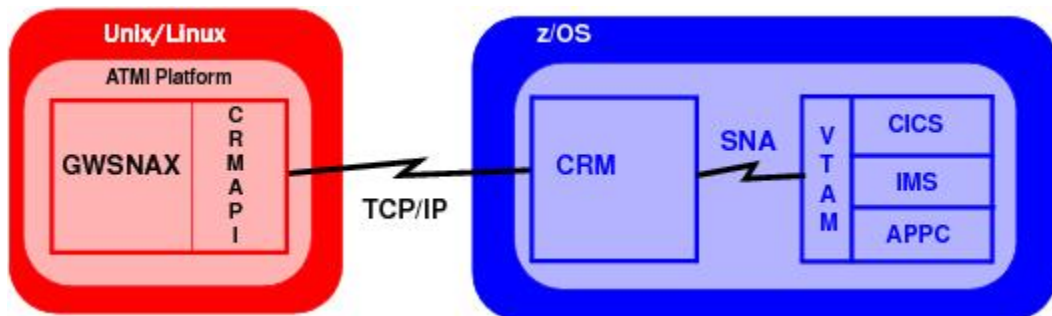
Figure 3-1 Local Oracle Tuxedo Mainframe Adapter for SNA Configuration on Unix/Linux Platform



3.1.1.2.2 Distributed Configuration—CRM on z/OS Host

This distributed configuration deploys the CRM to the z/OS host system as shown in the following figure. It employs TCP/IP connectivity with the host, eliminating the need for a local SNA stack. This configuration provides a faster network interface and is less complex than the local configuration.

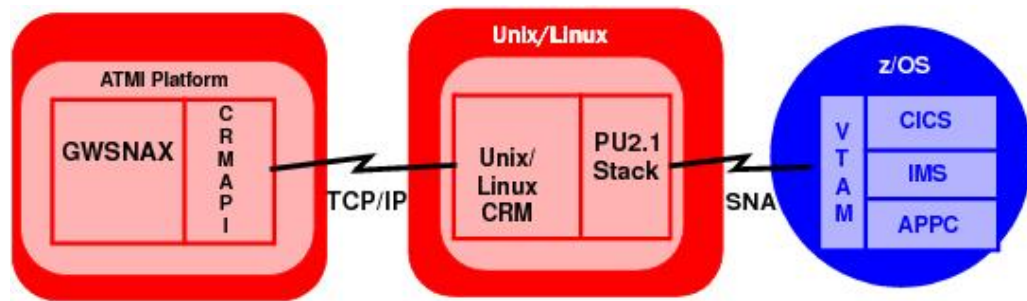
Figure 3-2 Distributed CRM on z/OS Platform



3.1.1.2.3 Distributed Configuration—CRM on Unix/Linux

This distributed configuration deploys the CRM and stack to a UNIX or Linux platform, as shown in the figure. It employs the TCP/IP connectivity between the Oracle Tuxedo Mainframe Adapter Gateway and CRM, as well as the SNA connectivity to the host. This configuration allows you to use multiple stacks from different stack vendors. Also, on the ATMI platform side, you have a greater variety of UNIX-based or Windows platform manufacturers to choose from.

Figure 3-3 Distributed CRM on UNIX/Linux Platform



3.1.2 Configure the Local Host

Ensure that the local host is prepared to conduct operations with the remote host by configuring the Local LU for the appropriate stack.

Refer to the Oracle Tuxedo Mainframe Adapter for SNA *CRM Administration Guide* for more information about this task.

3.1.3 Configure the Remote Host

Ensure that the remote host is prepared to conduct operations with the ATMI local domain by completing the following tasks:

1. Configure the Remote LU.
2. Complete cross-platform definitions, if necessary.
3. Activate the connection between the remote host and the local host.

Refer to the Oracle Tuxedo Mainframe Adapter for SNA *CRM Administration Guide* for more information about these tasks.

3.2 Configuring the Oracle Tuxedo Mainframe Adapter for SNA Gateway

- [Overview](#)
- [Configurations for Customized Call ID Support](#)
- [Overview](#)
- [Configurations for Customized Call ID Support](#)

3.2.1 Overview

The following list summarizes the tasks that must be completed to configure the Oracle Tuxedo Mainframe Adapter for SNA Gateway (**GWSNAX**):

1. Edit the **DMTYPE** file.
2. Edit the **UBBCONFIG** file and load to create the binary.

3. Edit the `DMCONFIG` file (and CRM configuration file, if you need separate CRM configurations on mainframe) and load `DMCONFIG` to create the binary.
4. Start the CRM.
5. Start the ATMI servers.

In addition, if you want to retrieve customized call ID from every Oracle Tuxedo Mainframe Adapter incoming call message, do the following.

- [Configurations for Customized Call ID Support](#)
- [Step 1: Edit the DMTYPE File](#)
- [Step 2: Edit the UBBCONFIG File](#)
- [Step 3: Edit the DMCONFIG File \(and Edit CRM Configuration File to Have Separate CRM Configurations on Mainframe\)](#)

3.2.1.1 Step 1: Edit the DMTYPE File

The `DMTYPE` file is an ASCII file. Use any text editor to edit this file.

1. Insert the following line in the `DMTYPE` file located in the `$TUXDIR/udataobj` directory:
For UNIX:

```
SNAX:::
```

For Windows:

```
SNAX;;;
```

2. Ensure that the `$TUXDIR/udataobj/DMTYPE` file exists prior to editing the `DMCONFIG` file. See `dmloadcf` in Appendix A, [Administration Mode Commands](#) for more information.

3.2.1.2 Step 2: Edit the UBBCONFIG File

The `UBBCONFIG` file is an ASCII file that can be edited with any text editor. To edit the `UBBCONFIG` file, complete the following tasks:

1. Create a `UBBCONFIG` file for each application. Refer to the Configuration section in the appropriate *ATMI platform product documentation* for more specific information about the `UBBCONFIG` file.
2. Establish a new gateway configuration or modify an existing one by defining the domain and gateway administrative servers to the ATMI system in the `UBBCONFIG` file.
3. If the CRM is to run as an ATMI server in a local configuration, add a CRM entry to the `*SERVERS` section of the `UBBCONFIG` file. For a description, refer to the Oracle Tuxedo Mainframe Adapter for SNA *CRM Administration Guide*.
Note: If the CRM is started as an ATMI process, it must precede the `GWSNAX` entry in the `UBBCONFIG` file.
4. Establish the Oracle Tuxedo Mainframe Adapter for SNA Gateway by adding an entry to the `*SERVERS` section of the `UBBCONFIG` file. For a description, refer to the `GWSNAX` command in Appendix A, [Administration Mode Commands](#). The following gateway features may be enabled in the `UBBCONFIG` file:
 - Data transformation

- Bypassing user ID mapping
 - Encryption
 - Authentication
5. Refer to the appropriate ATMI platform documentation for instruction for using `tmloadcf` to load the `UBBCONFIG` file.

Listing Sample UBBCONFIG File Entries Specifying CRM as an ATMI Server

```
*GROUPS
    SNAGRP  LMID=mysys
           GRPNO=4

    LOCGRP  LMID=mysys
           BRPNO=5
*SERVERS
    DEFAULT:CLOPT = "-A"

    DMADM  SRVGRP=LOCGRP
           SRVID=14

    GWADM  SRVGRP=SNAGRP
           SRVID=14
           REPLYQ=Y
           RESTART=N
           GRACE=0

    SNACRM SRVGRP=SNAGRP
           SRVID=15
           CLOPT="-A--//dalhps2:4452 SNAGRP"
           RESTART=Y
           RCMD=rstsnagrp
           GRACE=120
           MAXGEN=2

    GWSNAX SRVGRP=SNAGRP
           SRVID=16
           RQADDR="SNADOM"

    REPLYQ=N
    RESTART=Y
    RCMD=rstsnagrp
    GRACE=120
    MAXGEN=2
```

3.2.1.3 Step 3: Edit the DMCONFIG File (and Edit CRM Configuration File to Have Separate CRM Configurations on Mainframe)

There are two options to do this step; pick the one that best meets your requirements.

- [Option A: Edit the DMCONFIG File](#)
Follow the instructions in this part to complete this step, if you do not require separate CRM configurations on mainframe.
- [Option B: Edit the DMCONFIG File and CRM Configuration File](#)

Follow the instructions in this part to complete this step, if you require separate CRM configurations on mainframe.

- [Option A: Edit the DMCONFIG File](#)
- [Option B: Edit the DMCONFIG File and CRM Configuration File](#)

3.2.1.3.1 Option A: Edit the DMCONFIG File

The configuration specified in the `DMCONFIG` file controls much of the operation of the Oracle Tuxedo Mainframe Adapter for SNA Gateway (GWSNAX). A sample of this file is provided in the installation directory of your Oracle Tuxedo Mainframe Adapter for SNA product software.

Note:

Because Oracle Tuxedo Mainframe Adapter for SNA may be installed on a variety of platforms, the procedures in this section make only general references to command entries. Many steps show UNIX command examples. Be sure to use the proper syntax for your platform when making command-line entries.

1. Verify that the Oracle Tuxedo Mainframe Adapter for SNA product software is installed and accessible to your text editor.
2. Verify that you have file permission to access the install directory and modify the sample `DMCONFIG` file.
3. Set each of the parameters of the `DMCONFIG` file as described in the following sections and load the `DMCONFIG` file. Refer to the appropriate ATMI documentation for instruction for using `dmloadcf` to load the `DMCONFIG` file.

- a. Update the `*DM_LOCAL_DOMAINS` Section

This section identifies local domains and their associated gateway groups. The section must have an entry for each gateway group (Local Domain). Entries have the form:

```
LDOM required parameters {optional parameters}
```

In this entry, `LDOM` is an identifier value used to name each local domain. For a full description of the optional and required parameters, see `DMCONFIG` in Appendix A, [Administration Mode Commands](#).

For each `LDOM` entry, the value of the `TYPE` parameter distinguishes this gateway from other gateway types. Currently, `SNAX` replaces the value `SNADOM` used in previous releases. The parameter entry takes the form:

```
TYPE={SNAX | OSITP | TDOMAIN}
```

Select the value `TYPE=SNAX` for the `LDOM` entry.

To support `SNAX` reconnects to CRM automatically (as CRM restarting), you can set `CONNECTION_POLICY` to `ON_STARTUP` and specify a `RETRY_INTERVAL` (the default value is 60 seconds). For example:

```
CONNECTION_POLICY=ON_STARTUP  
RETRY_INTERVAL=60
```


b. Update the *DM_REMOTE_DOMAINS Section

This section identifies the known set of remote domains and their characteristics. Entries have the form: RDOM required parameters

In this entry, RDOM is an identifier value used to identify each remote domain known to this configuration. For a full description of the required parameters, see DMCONFIG in Appendix A, [Administration Mode Commands](#) .

For each RDOM entry, the value of the TYPE parameter indicates that the remote domain communicates using the SNA protocol. The parameter entry takes the form:

```
TYPE={SNAX | OSITP | TDOMAIN}
```

Select the value TYPE=SNAX for the RDOM entry.

c. Add the *DM_SNACRM Section **Note:**

*DM_SNACRM, *DM_SNASTACKS, and the *DM_SNALINKS sections have replaced the *DM_SNADOM section used in previous releases of eLink Adapter for Mainframe.

Any changes to the *DM_SNACRM, *DM_SNASTACKS, or *DM_SNALINKS sections require a cold start for the Oracle Tuxedo Mainframe Adapter for SNA domain. If you do not cold start the Oracle Tuxedo Mainframe Adapter for SNA domain, an error will occur on domain start-up indicating cold start required for the configuration change.

The *DM_SNACRM section provides three keywords used to identify the CRM that provides ATMI transaction semantics in a given domain and its partners. Entries have the general form:

```
<CRMName> parameters
```

In this entry, <CRMName> is the locally known name of this SNACRM definition to be used when referencing this SNACRM in subsequent sections. This name is an ASCII string 1-30 characters in length. The parameters are the keyword/value pairs that make up the definition. All keywords are required for a valid SNACRM definition. Keywords can be in any order.

```
LDOM= <LocalDomainName> (Required)
```

LDOM associates this SNACRM with a defined local domain. <LocalDomainName> is the reference to an entry in the *DM_LOCAL_DOMAINS section. This name is an ASCII string 1 to 30 characters in length. This parameter is required. This parameter has no default

```
SNACRMADDR= <HexSocketAddress> (Required)
```

SNACRMADDR provides the socket address the domain gateway uses to communicate with the SNACRM. This address represents the machine and port where the CRM runs. In a local configuration, this address is the local platform. In a distributed configuration, this address is a remote platform. This

address must be used on the `SNACRM` command line. This parameter is required and has no default.

`<HexSocketAddress>` is a TCP/IP address using `/hostname:port_addr` or the `sockaddr_in` format of family, port, address:

```
<0xFFFFPPPPAAAAAAAA>
```

In this entry, arguments and options are defined in the following way:

`FFFF` is the hex value of the protocol family, always `0x0002` for the INET family.

`PPPP` is the hex value of an unused TCP/IP port.

`AAAAAAAA` is the hex value of the IP address for the machine running the `SNACRM`. Therefore, if the CRM was running on a machine named `myhost` with an IP address of `206.189.43.13`, and you wanted to use port `6000` for the CRM, then `SNACRMADDR` would be: `//myhost:6000` or `0x00021770CEBD2B0DNWDEVICE= <DeviceName>` (Required) `<Device Name>` is the logical name used to access the network. For example: `/dev/tcp`

d. Add the `*DM_SNASTACKS` Section

The `DM_SNASTACKS` section provides five keywords that identify the third-party SNA stack that must be used for connections established between a given domain and its partners. Entries have the general form:

```
<StackReference> parameters
```

In this entry, `<StackReference>` is the locally known name of this stack definition and it is used when referencing this stack in subsequent sections. This name is an ASCII string 1-30 characters in length. The parameters are the keyword/value pairs that make up the definition. Keywords can be in any order. All keywords are required for a valid stack definition.

```
LOCALLU= <LocalLUAlias> (Required)
```

`LOCALLU` provides a reference to an LU alias defined in the third-party SNA stack. `<LocalLUAlias>` is the name used to identify the local LU definition as specified by the third-party SNA stack configuration. This name represents the end node for an LU6.2 connection. The value for this parameter is an ASCII string, 1-8 characters in length. This parameter is required. This parameter has no default. The third party SNA stack requires a corresponding definition for a local LU.

```
LTPNAME= <LocalTransactionProgramName> (Required)
```

`LTPNAME` identifies the inbound transaction programs that are serviced by any `SNACRM` using this stack definition. `<LocalTransactionProgramName>` is the name used to identify inbound transaction programs for which an attach will be accepted. The only useful value is an asterisk that indicates all inbound attaches will be accepted. This parameter is required. This parameter has no default. Partial TP names are not supported. The third-party SNA stack requires a corresponding definition for inbound TP names.

```
SNACRM= <CRMName> (Required)
```

`SNACRM` provides a name to which the associated `SNACRM` definition is referenced. `<CRMName>` is the name used to associate the `*DM_SNACRM` definition with this `*DM_SNASTACKS` entry. The value for this parameter is an ASCII string, 1-30 characters in length. This parameter is required. This parameter has no default.

STACKPARMS= <parameters required for third-party sna stack>
(Required)

STACKPARMS provides a method for the domain gateway to pass any required parameters to the third party SNA stack. The <parameters required for third-party sna stack> is an ASCII string, 1-128 characters in length. Currently, the only value used is the TCP/IP hostname for the machine running the third-party SNA stack. This parameter is required. This parameter has no default.

STACKTYPE={VT210 | IBM60}

This option is used to indicate which vendor SNA stack is being used. It is also used to determine the names of specific Oracle Tuxedo Mainframe Adapter for SNA system libraries. Because of this, it is essential that the value of this option be coded correctly. These values are mapped to the equivalent Oracle Tuxedo Mainframe Adapter for SNA system library.

e. Add the *DM_SNALINKS Section

The *DM_SNALINKS section provides 11 key words that define the SNA Link information required by domains of type SNA. Entries have the general form:<Link Name> parameters

In this entry, <Link Name> is the identifier value used to identify the connection between a local domain (LDM) and a remote domain (RDM). This name is an ASCII string 1-30 characters in length. The parameters are the keyword/value pairs that make up the definition. Keywords can be in any order.

STACKREF= <Stack Reference> (Required)

This required parameter defines the stack that will be used for establishment of this link. The STACKREF string is the tag that was used in the corresponding definition established in the *DM_SNASTACKS section.

RDM= <name>

Each link defines a connection between an ATMI system application domain and a remote system connected with an SNA network. The remote system is, in ATMI terms, a remote domain. The RDM option associates the link with a remote domain. This remote domain must have been configured with the TYPE=SNAX option. The RDM name must match an RDM value previously identified in the *DM_REMOTE_DOMAINS section.

LSYSID= <name>

LSYSID is the four-character identifier for this link. This must match the connection ID in the CICS/ESA resource definition used by a partner CICS/ESA to communicate to the SNACRM across this link. If you are using the macro definition, it is a four-character name on the SYSIDNT option of the DFHTCT macro.

RSYSID= <name>RSYSID is the four-character remote sysid of the partner. Typically it is the sysid of a CICS/ESA region, but could also be the subsystem ID of an IMS control region. This parameter must match the actual sysid of the remote partner. This name is the SYSIDNT of the DFHSIT or the value in the CICS/ESA start-up overrides

RLUNAME= <name> (Required) The RLUNAME value represents an alias known to the third-party SNA stack that resolves to a VTAM netname for the remote

application. This remote application is most likely the VTAM `applid` for a CICS/ESA region, however it could also be an APPC/MVS LU defined for use with IMS. The value must be unique within the SNA network. The value *name* must be 1-8 characters. This parameter is required. This parameter has no default. The third-party stack configuration requires a matching definition. `MODENAME= <name>`

(Required)`MODENAME` is VTAM mode entry defined to the third-party SNA stack. For a CICS/ESA link, this entry must be compatible with the session definition or profile entry for the corresponding connection. For an IMS connection, this entry must be compatible with the `DLOGMOD` entry on the LU definition used to access the IMS scheduler. The value *name* must be 1-8 ASCII characters. This parameter is required. This parameter must match the third-party SNA stack configuration and must be compatible with the corresponding entries defined to VTAM and/or CICS/ESA.

`SECURITY={LOCAL | IDENTIFY | VERIFY | PERSISTENT | MIXIDPE}SECURITY`

specifies the security setting in CICS/ESA connection resource definition. It identifies the level of security enforced under CICS/ESA by the external security manager. Legal values are `LOCAL`, `IDENTIFY`, `VERIFY`, `PERSISTENT` or `MIXIDPE`. The default setting is `LOCAL`. `PERSISTENT` and `MIXIDPE` identify the setting in the remote connection definition, but are identical to the `VERIFY` option in this release of Oracle Tuxedo Mainframe Adapter for SNA.`MAXSESS= <number>`

This number represents the maximum number of sessions that can be concurrently acquired on this link. It must be greater than or equal to four, and less than or equal to the maximum number of sessions that can be configured by the SNA stack. The actual number of concurrent sessions is determined by both system configurations to be the lowest maximum number of sessions allowed by either system. The default value is `MAXSESS=64`.

`MINWIN= <number>`

This value is the minimum number of contention winners. Typically, this value is half the `MAXSESS` value. This number added to all CICS/ESA session definition winner numbers for the connection must be equal to the `MAXSESS` value. The default value is `MINWIN=0`.`STARTTYPE={AUTO|COLD}`

This option sets the recovery mode for transactional links. When set to `AUTO`, the system restarts using configuration and link data recovered from the transaction log. When set to `COLD`, the system uses configuration data taken from the current `DMCONFIG` file and loses any in-flight link data. Changing `DMCONFIG` file parameters and performing an `AUTO` start results in a message warning that changed parameters are ignored until the next cold start. `MAXSYNCLVL={0 | 1 | 2}`

This value represents the maximum sync-level conversation that can be supported on this link. The default is sync-level 2. Transaction support is only available at sync-level 2. `Sync-level 0`

A value of zero (0) means this link is non-transactional. No synchronization is maintained. This value can be used for sending and receiving messages from IMS via the APPC/MVS transparency interface. `Sync-level 1`

Allows sync-level 0 capabilities as well as support for `SYNCONRETURN Distributed Program Link (DPL)` with CICS/ESA systems (outbound ATMI `tpcall()` requests with `TPNOTRAN`). `Sync-level 2`

Supports all sync-level 0 and sync-level 1 features for systems able to exchange logs and compare states. In addition, full syncpoint synchronization at sync-level 2 is supported.

f. Update the *DM_LOCAL_SERVICES Section

The *DM_LOCAL_SERVICES section provides information on the services exported by each local domain. Entries have the general form: <Local Service Name> parameters

In this entry, <Local Service Name> is the local name of the exported service. This name is an ASCII string 1-15 characters in length. The parameters are the keyword/value pairs that make up the definition. Keywords can be in any order. For a full description of parameters, see DMCONFIG in Appendix A, [Administration Mode Commands](#). RNAME= <name> (Required)

The RNAME option is the local-service name imported from a remote CICS/ESA region. This name is used by the CRM to select a local service.

When the RNAME specifies an alternate mirror transaction identifier for explicit attachment for inbound DPL requests, it must be a combination of the alternate mirror TRANSID and a CICS/ESA program name in the following format: RNAME=AAAA:BBBBBBBB

In this statement, the arguments and options are defined in the following way: AAAA is a 1-4 character alternate mirror TRANSID. BBBBBBBB is a 1-8 character CICS/ESA program name. The colon is required to indicate the TRANSID/program name combination. The TRANSID must be composed of acceptable CICS/ESA characters:

```
A-Za-z0-9$@#./-_%&Q?!"|=, ; <>
```

Refer to “Special Treatment of TRANS ID for DPL” in “Application to Application Programming Considerations.”

g. Update *DM_REMOTE_SERVICES Section

The *DM_REMOTE_SERVICES section provides information on services “imported” and available on remote domains. Entries have the general form: <Remote Service Name> parameters

In this entry, <Remote Service Name> is the name used by the local application for a particular remote service. This name is an ASCII string 1-15 characters in length. The parameters are the keyword/value pairs that make up the definition. Keywords can be in any order. For a full description of parameters, see DMCONFIG in Appendix A, [Administration Mode Commands](#).

```
FUNCTION={APPC | DPL}
```

The FUNCTION option has been added to allow outbound ATMI service requests to map to APPC transaction programs or CICS/ESA DPL programs. The default value is APPC. RNAME= <name> The RNAME option is the name of the host TP_NAME. For non-CICS/ESA systems, this name can be up to 64 characters in length. For CICS/ESA systems, this name is the transaction ID for FUNCTION=APPC and the program name for FUNCTION=DPL requests. CICS/ESA trans-id names cannot exceed four characters and CICS/ESA program names cannot exceed eight characters. The RNAME option must observe these requirements.

When the `RNAME` specifies an alternate mirror transaction identifier for explicit attachment to outbound DPL requests, it must be a combination of the alternate mirror `TRANSID` and an advertised remote CICS/ESA program name in the following format: `RNAME=AAAA:BBBBBBBB`

In this statement, the arguments and options are defined in the following way: `AAAA` is a 1-4 character alternate mirror `TRANSID`. `BBBBBBBB` is a 1-8 character CICS/ESA program name. The colon is required to indicate the `TRANSID`/program name combination. The `TRANSID` must be composed of acceptable CICS/ESA characters:

```
A-Za-z0-9$@#./-_%&Q?!"'=" ; <>
```

Refer to “Special Treatment of TRANS ID for DPL” in the “Application to Application Programming Considerations” section of the Oracle Tuxedo Mainframe Adapter for SNA *Reference Guide*.

3.2.1.3.2 Option B: Edit the DMCONFIG File and CRM Configuration File

Oracle Tuxedo Mainframe Adapter for SNA 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`.

Perform the following steps on both SNA side and CRM side to use the flexibility.

- On SNA Side
On this side, do all steps listed in [Option A: Edit the DMCONFIG File](#), except for its Add the `*DM_SNASTACKS` Section and Add the `*DM_SNALINKS` Section.
- On CRM Side
On this side, do the steps listed in [Having Separate CRM Configurations on Mainframe](#) in *Oracle Tuxedo Mainframe Adapter for SNA CRM Administration Guide*.

3.2.2 Configurations for Customized Call ID Support

You can retrieve customized call ID from every Oracle Tuxedo Mainframe Adapter incoming call message. Many buffer types support this feature, including `CHANNEL/CONTAINER`, `VIEW`, and `FML32`. Call ID is composed of fields; you can choose to put one or more fields to each call ID in configuration file. Oracle Tuxedo Mainframe Adapter for SNA Gateway loads and parses the configuration file at initialization, and generates a call ID depending on it when decoding an inbound request and sets this call ID to message.

The maximum length of call ID is 64 bytes. Excessive characters will be trimmed. Trailing spaces will be trimmed no matter what the current length is.

You can specify each field's length. The maximum length of field is 64 bytes. Excessive characters will be trimmed. Spaces will be padded if the actual length is shorter than specified.

If the environment variable `CALLID_FIELDS_CFG` and call ID configuration file are correctly set for Oracle Tuxedo Mainframe Adapter for SNA Gateway, and Oracle Tuxedo System and Applications Monitor Plus (TSAM Plus) call path is enabled on SNA Gateway, the call ID is

collected to the ECID field in the TSAM Plus call path. For more information, see [Call Path](#) in *Oracle Tuxedo System and Applications Monitor Plus Console Users Guide*.

To utilize this support, do the following steps.

1. [Set Environment Variable CALLID_FIELDS_CFG](#) to enable this support, specifying configuration file location and field names.
2. [Set Configuration File CALLFLDS](#).

For a sample of configurations for customized call ID, see the [Sample](#).

 **Note:**

- This feature requires TSAM Plus(12.1.3.0 and above).
- This feature only supports `STRING` and `CARRAY` field types in `VIEW`. If the field type is `CARRAY`, TMA represents it in hexadecimal characters in upper case and doubles its length (two characters represent one single byte).

- [Set Environment Variable CALLID_FIELDS_CFG](#)
- [Set Configuration File CALLFLDS](#)
- [Sample](#)

3.2.2.1 Set Environment Variable CALLID_FIELDS_CFG

Set `CALLID_FIELDS_CFG` to enable this support. If it is not set, this support will not be enabled. `CALLID_FIELDS_CFG` also specifies the configuration file location and field names. Both absolute file path and relative file path are supported.

```
export CALLID_FIELDS_CFG=/APPDIR/callid.ini
```

3.2.2.2 Set Configuration File CALLFLDS

- [Syntax](#)
- [Descriptions](#)

3.2.2.2.1 Syntax

```
[<Service_Name>]
CALLFLDS = <View_Name_1>: <Field_Name_1>: <Offset_begin_1>-
<Offset_end_1> [, ..., <View_Name_n>:
    <Field_Name_n>: <Offset_begin_n>-<Offset_end_n>]
```


3.2.2.2 Descriptions

Table 3-1 CALLFLDS Parameters

Parameter	Description
<Service_Name>	Section name (service name). This name must match the local service name defined in DM_LOCAL_SERVICES section of DMCONFIG. For nested VIEW, this name is the closest VIEW name which includes this field. For channel/container, because container name is mapped to VIEW name, those two names are the same.
<View_Name_n>	VIEW name defined in VIEW file. It must be the same as the name defined in OUTBUFTYPE of DM_LOCAL_SERVICES section in DMCONFIG; otherwise, though no errors are reported, the configuration is invalid.
<Field_Name_n>	Field name of VIEW. It must be the same name as cname in VIEW file; otherwise, though no errors are reported, the configuration is invalid.
<Offset_begin_n> - <Offset_end_n>	Lower and higher positions in call ID string. Positions must be positive numbers and must be separated by "-".



Note:

- You can define multiple service sections and multiple fields.
- Every separator is required. If the total number of separator ":" is not equal to 2 in any one of the field definitions, errors are reported.

3.2.2.3 Sample

In this scenario, Tuxedo local services defined in DMCONFIG are configured as follows. OUTBUFTYPE here defines the message format of the incoming call.

Listing Scenario: Tuxedo Local Services Defined in DMCONFIG

```
[REVIEW]
CONV=N
RNAME="REVIEW"
INBUFTYPE="VIEW32:flat1"
OUTBUFTYPE="VIEW32:nest"
```

And nested VIEW is defined in bufview.v as follows.

Listing Scenario: Nested VIEW Defined in bufview.v

```
#type    cname    ffname    count    flag    size    null
VIEW     ECvw
carray   Eca      ECA       1        -       20     -
END
```



```

VIEW      EDvw
string    EDCstr    EDCSTR    1      -      -      -
END

VIEW      Evw
string    ECstr     ECSTR     1      -      -      -
short    EBsh      EBSH     1      -      -      -
struct   ECvw      ECVW     1      -      -      -
struct   EDvw      EDVW     1      -      -      -
END

VIEW nest
char      Ach       ACH       1      -      -      -
short    Bsh       BSH       1      -      -      -
string    Cstr      CSTR      1      -      41     -
long     Dlo       DLO       1      -      -      -
struct   Evw       EVW       1      -      -      -

END

```

Then you must set environment variable as follows.

```
export CALLID_FIELDS_CFG=/APPDIR/callid.ini
```

And define configuration file as follows.

```
[REVIEW]
CALLFLDS= nest: Cstr:0-2, Evw: ECstr :3-7, EDvw: EDCstr:8-15, ECvw:
Eca:16-63
```

3.3 Using the Oracle Tuxedo Metadata Repository

TMA SNA Gateway imported and exported services are configured in the `DMCONFIG` file `DM_REMOTE_SERVICES` and `DM_LOCAL_SERVICES` sections. The service configuration includes both service definition and service deployment.

The following service parameters in the `DMCONFIG` file `DM_REMOTE_SERVICES` and `DM_LOCAL_SERVICES` sections are service definition parameters:

- CONV
- INBUFTYPE
- OUTBUFTYPE
- FUNCTION

Meanwhile the Oracle Tuxedo Metadata Repository provides a centralized repository for Oracle Tuxedo service definitions. You can either configure the service definitions for TMA SNA in the `DMCONFIG` file or in the Oracle Tuxedo Metadata Repository.

- [Tuxedo Metadata Respository Service-Level Keywords](#)

- [Mapping Between DMCONFIG Elements and Tuxedo Metadata Repository Keywords](#)
- [GWSNAX Server Configuration](#)
- [Tuxedo Metadata Repository Service-Level Keywords](#)
- [Mapping Between DMCONFIG Elements and Tuxedo Metadata Repository Keywords](#)
- [GWSNAX Server Configuration](#)

3.3.1 Tuxedo Metadata Repository Service-Level Keywords

In order for TMA SNA to use the service definition in the Oracle Tuxedo Metadata Repository, you must configure the following Oracle Tuxedo Metadata Repository service-level keywords for the corresponding TMA SNA services:

- `servicemode`
- `inbuf`
- `outbuf`
- `servicetype`
- `SNAISC`
- `inview`
- `outview`

For more information, see [Using Service-Level Keywords and Values](#) in *Managing The Oracle Tuxedo Service Metadata Repository*.

3.3.2 Mapping Between DMCONFIG Elements and Tuxedo Metadata Repository Keywords

The following DMCONFIG elements must be mapped to Oracle Tuxedo Metadata Repository Keywords.

- `CONV = { Y | N }` (defined in `DM_LOCAL_SERVICES` and `DM_REMOTE_SERVICES` sections)
Re-use the keyword `servicetype` in metadata repository to store `CONV`. You need to use `servicetype=conv` to map `CONV=Y` element in DMCONFIG; otherwise, `CONV=N` element is mapped.
- `INBUFTYPE = type[:subtype]` (defined in `DM_LOCAL_SERVICES` and `DM_REMOTE_SERVICES` sections)
Use keyword `inbuf` to store buffer type; use `inview` to store buffer subtype for input buffer (Optional).
- `OUTBUFTYPE = type[:subtype]` (defined in `DM_LOCAL_SERVICES` and `DM_REMOTE_SERVICES` sections)
Use keyword `outbuf` to store buffer type; use `outview` to store buffer subtype for output buffer (Optional).
- `FUNCTION = {APPC|DPL}` (only defined in `DM_REMOTE_SERVICES` section)
Use keyword `snaisc` to store `FUNCTION`'s value. It is only available for remote services and is not presented in local service.

Additionally, for services defined in `DM_LOCAL_SERVICES` section, set `servicemode` value to `tuxedo`; for services defined in `DM_REMOTE_SERVICES` section, set `servicemode` value to `sna`.

3.3.3 GWSNAX Server Configuration

To use the Oracle Tuxedo Metadata repository, you must configure the [GWSNAX](#) server with the `-M` parameter.

 **Note:**

The Oracle Tuxedo Metadata repository server, [TMMETADATA \(5\)](#), must be configured, and must be started before the [GWSNAX](#) server.

3.4 Starting the System

To start your Oracle Tuxedo Mainframe Adapter for SNA system, you must first start the CRM and then start the ATMI Servers as described in the following sections.

- [Step 1: Start the CRM](#)
- [Step 2: Start the ATMI Servers](#)

3.4.1 Step 1: Start the CRM

If the CRM is run in distributed mode or from the command line, it must be started independently of the ATMI processes. Start the CRM in one of the following ways:

- For MVS, use the JCL that was modified during the installation process.
- For all other distributed configurations, use the `CRM` command in the following format:

```
CRM [parameters] <HexSocketAddress> <group name>
```

Refer to `CRM` in Appendix A, [Administration Mode Commands](#) for more information about this command.

3.4.2 Step 2: Start the ATMI Servers

Perform a `tmboot` as described in the appropriate ATMI platform documentation to start the ATMI servers. If it is already running, perform a `tmshutdown` and `tmboot`.

 **See Also:**

- [Administration Mode Commands](#)
- [Managing The Oracle Tuxedo Service Metadata Repository](#)
- [TMMETADATA \(5\) and UBBCONFIG 5 in Section 5 - File Formats, Data Descriptions, MIBs, and System Processes Reference](#)

4

Verifying the Software

After installing and configuring the Oracle Tuxedo Mainframe Adapter for SNA software, verify the operational integrity of the environment by running a sample application on a simple server in client/server transaction scenarios. This process employs programs available in your product software libraries.



Note:

All references to ATMI files, functions, and documentation apply to Tuxedo files, functions and documentation.

The sample applications are located in the simple server library (`TMA/sna/simpapp`). The simple server passes a string from the client to the server.

The CICS/ESA programs may run as either DTP or DPL processes, and as either servers or clients. The simple server may also run in either transactional or non-transactional mode. In the transactional mode, these scenarios verify that the sync-level 2 protocol is established between the two application environments.

When the client runs as an ATMI client, the server runs as a CICS/ESA host. Enter a text string in lower-case letters with command arguments. The CICS/ESA server converts the lower-case letters to upper-case letters and re-displays the text string.

When the client runs as a CICS/ESA client, the server runs as an ATMI server. Again, enter a text string in lower-case letters. The ATMI server converts the text string into a mirror image and displays the string as reversed letters.



Note:

The verification process is intended for the CICS/ESA environment only between ATMI applications and CICS/ESA applications. If your ATMI applications operate in other environments, you must create your own verification process. See the [APPC/IMS Programming](#) section.

This chapter contains the following topics:

- [Building Verification Tests](#)
- [Running the Sample Application](#)

4.1 Building Verification Tests

Build the verification test to run in two domains, the ATMI local domain and the CICS/ESA remote domain. The executables in each domain are different. The following sections discuss how to build these executables.

- [Building ATMI Platform Executables](#)
- [Building CICS/ESA Executables](#)

4.1.1 Building ATMI Platform Executables

To build the ATMI platform executables, perform the tasks that are described in detail in the following sections:

- [Modify the UBBCONFIG File](#)
- [Execute the tmloadcf Command](#)
- [Modify the DMCONFIG File](#)
- [Execute the dmloadcf Command](#)
- [Modify the Environment Files](#)
- [Build the Server](#)
- [Build the Client](#)
- [Modify the UBBCONFIG File](#)
- [Execute the tmloadcf Command](#)
- [Modify the DMCONFIG File](#)
- [Execute the dmloadcf Command](#)
- [Modify the Environment Files](#)
- [Build the Server](#)
- [Build the Client](#)

4.1.1.1 Modify the UBBCONFIG File

Make the following modifications to the `UBBCONFIG` file.

- [Add the ATMI Simple Server to the UBBCONFIG File](#)
- [Enable Transactional Services](#)

4.1.1.1.1 Add the ATMI Simple Server to the UBBCONFIG File

Modify the `UBBCONFIG` file to include the name of the ATMI simple server in the following way:

```
GROUPS
GROUP3          LMID=sna GRPNO=3

SERVERS
mirrorsrv      SRVGRP=GROUP3 SRVID=1 RQADDR=MIRR1 REPLYQ=Y

SERVICES
MIRROR
```

4.1.1.1.2 Enable Transactional Services

If you plan to run the transactional version of the verification process, enable the `TLOGDEVICE` comment in the Machine section to point to a valid DTP transaction log. To run transaction examples, create the DTP transaction log named on the `UBBCONFIG` `TLOGDEVICE` with the ATMI platform bulletin modification interpreter `tmadmin`. Refer to the appropriate ATMI *platform documentation* for option descriptions.

The application server `GROUP3` in the Groups section must point to a valid transaction manager server. For example:

```
GROUPS
GROUP3          LMID=sna GRPNO=3 TMSNAME=tsttms TMSCOUNT=2

SERVERS
mirrorsrv      SRVGRP=GROUP3 SRVID=1 RQADDR=MIRR1 REPLYQ=Y

SERVICES
MIRROR
```

4.1.1.2 Execute the `tmloadcf` Command

Execute the ATMI `tmloadcf` command to parse the `UBBCONFIG` file and create a binary version of the file. Refer to the appropriate ATMI *platform documentation* for option descriptions.

For example:

```
tmloadcf UBBCONFIG
```

Respond to the prompts as the command executes.

4.1.1.3 Modify the `DMCONFIG` File

The `DMCONFIG` file must contain both local and remote definitions for the simple server.

Note: A sample `DMCONFIG` file is included with the simple server.

Listing Sample `DMCONFIG` File

```
*DM_LOCAL_SERVICES
#The ATMI reverse string server
MIRROR          LDOM="simpsnad"
                CONV=N
                RNAME="MIRRORSERV"
                INBUFTYPE="STRING"
                OUTBUFTYPE="STRING"
DOUBLEMIRROR
                CONV=N
                RNAME="MIRRDPLS"
                INBUFTYPE="STRING"
                OUTBUFTYPE="STRING"
*DM_REMOTE_SERVICES
#The CICS upper-case DTP and DPL servers
```

```
SIMPDPL      AUTOTRAN=N
             LDOM="simpsnad"
             RDOM=SNAG1
             CONV=N
             RNAME="TOUPDPLS"
             INBUFTYPE="STRING"
             OUTPBUFTYPE="STRING"
             FUNCTION="DPL"
SIMPDTP      AUTOTRAN=N
             LDOM="simpsnad"
             RDOM=SNAG1
             CONV=N
             RNAME="DTPS"
             INBUFTYPE="STRING"
             OUTPBUFTYPE="STRING"
             FUNCTION="APPC"
```

In the preceding `DMCONFIG` file example, both instances of the `LDOM` name correspond to the SNA domain name in the `DM_LOCAL_DOMAINS` section. The server is a request/response server.

In the `DM_LOCAL_SERVICES` section, the `RNAME="MIRRORSERV"` and `RNAME="MIRRDPLS"` values are the names passed from the CICS/ESA environment. `MIRROR` and `DOUBLEMIRROR` refer to the advertised services provided by the `mirrorsrv` server named in the `UBBCONFIG` file. The `CONV=N` definition indicates the protocol that is observed by the SNA domain, although the CICS/ESA client does not perform an ATMI `tpcall`.

In the `DM_REMOTE_SERVICES` section, the `RNAME` value identifies what is invoked in the CICS/ESA domain. For the *Distributed Program Link (DPL)* request, the `RNAME` equals the name of the *program* called (or for alternate mirror transaction identifiers, a `TRANSID/program name` combination). For the *Distributed Transaction Processing (DTP)* request, the `RNAME` equals the name of the *transaction ID*.

If you want to run transactional verification tests, you must enter a link definition `MAXSYNCLVL=2` in the `DM_SNALINKS` section. If you want to run non-transactional DPL tests only, you must enter `MAXSYNCLVL=1`.

4.1.1.4 Execute the `dmloadcf` Command

Execute the ATMI `dmloadcf` command to parse the `DMCONFIG` file and create a binary version of the file. Refer to the appropriate ATMI *platform documentation* for option descriptions.

For example:

```
dmloadcf DMCONFIG
```

Respond to the prompts as the command executes.

4.1.1.5 Modify the Environment Files

Two types of files are provided with your Oracle Tuxedo Mainframe Adapter for SNA product software that can be used to define the application and/or machine

environments for verification testing. If their equivalents do not already exist, modify the files provided and make them available to your system. The files are `apps.env` and `<machine>.env`.

- [The `app.env` File](#)
- [The `<Machine>.env` Files](#)

4.1.1.5.1 The `app.env` File

Modify the `app.env` file and include it with the `ENVFILE` parameter in the `MACHINES` section of the `UBBCONFIG` file. The following listing is an example of the `app.env` file.

Listing The `app.env` File

```
app.env
APPDIR=<Your application directory here>
TUXCONFIG=<Your Tux configuration here>
BDMCONFIG=<Your Domain configuration here>
TUXDIR=<Your ATMI directory here>
```

4.1.1.5.2 The `<Machine>.env` Files

Modify the `<machine>.env` file that is appropriate for your system:

- `solaris.env`
- `ENV (MVS)`
- `aix.env`
- `hpux.env`
- `linux.env`
- `ntenv.bat`

Each of these files is executable. Once you have modified the appropriate file for your system, execute it to export the machine environment variables. Examples of some of the `.env` files are shown in the following listings.

The following listing is an example of a UNIX `.env` file.

Listing The `linux.env` File

```
linux.env
export APPDIR=<Your application directory bin here>
export TUXCONFIG=<Your ATMI configuration qualified name here>
export BDMCONFIG=<Your ATMI domain configuration qualified name here>
export TUXDIR=<Your ATMI product directory here>
export STACK=<Your stack product library here>
#example STACK=/opt/ibm/sna/lib64
export PATH=$APPDIR:$TUXDIR/bin:$PATH
export LD_LIBRARY_PATH=$TUXDIR/lib:$LD_LIBRARY_PATH:$STACK
```

The following listing is an example of an MVS `.env` file.

Listing The mvs.env File

```
*=====
* mvs.env
* Environment macros for running the CRM on MVS.
*
*
*-----
-
APPDIR=<Your site high-level qualifier here>
*example APPDIR=ORACLE
```

4.1.1.6 Build the Server

Use the ATMI platform `buildserver` utility to build the `mirrorsrv` server load module from the provided source file `mirrorsrv.c`. The source file contains two service entries, `MIRROR` and `DOUBLEMIRROR`, which will be advertised by the `mirrorsrv` server.

When executed, the `MIRROR` service receives a text string from the client, reverses the letters, and displays a mirror image of the input text string.

When executed, the `DOUBLEMIRROR` service receives a text string from the client, reverses the letters, and concatenates the reversed string to the forward image of the string.

The following command is an example of a command entry to invoke the `buildserver` utility:

```
buildserver -o mirrorsrv -f mirrorsrv.c -s MIRROR,DOUBLEMIRROR
```

Refer to the appropriate ATMI *platform documentation* for option descriptions.

4.1.1.7 Build the Client

Use the ATMI platform `buildclient` utility to build a client load module from the provided source file `toupclt`. When executed, the load module sends a lower-case text string to the server, which converts it to uppercase in several modes, causing different server scenarios to execute.

For example:

```
buildclient -o toupclt -f toupclt.c
```

Refer to the appropriate ATMI *platform documentation* for option descriptions.

4.1.2 Building CICS/ESA Executables

To build the CICS/ESA executables, perform the tasks that are described in detail in the following sections:

- [Choose the Source Code Language](#)
- [Transfer the Source Code to the Host](#)

- [Translate CICS/ESA Verbs](#)
- [COBOL Language Translator Example](#)
- [Compile the Translated Source File](#)
- [Create the Executable Object](#)
- [Running the Sample Application](#)
- [Choose the Source Code Language](#)
- [Transfer the Source Code to the Host](#)
- [Translate CICS/ESA Verbs](#)
- [Compile the Translated Source File](#)
- [Create the Executable Object](#)
- [Configure the CICS/ESA Application](#)
- [View Connection and Session Status](#)

4.1.2.1 Choose the Source Code Language

The CICS/ESA sample programs used for verification are unloaded during the installation of your Oracle Tuxedo Mainframe Adapter product software. These programs are available in two languages, COBOL and C. You must choose which language to use to build the CICS/ESA executable object code. (Your choice might be affected by the type of compiler available on your MVS host.)

You can identify the sample program names by their suffixes:

- `TOUPDTPS.c` is the C language program name
- `TOUPDTPS.cbl` is the COBOL language program name

Although the structures of the sample programs are different, they both perform the same function. The ATMI platform executable program that you build communicates with either.

4.1.2.2 Transfer the Source Code to the Host

Transfer the source code to the host by the method you prefer, for example FTP (File Transfer Program). The destination could be a sequential dataset or a PDS file. The following table lists the source code files provided for UNIX and MVS platforms. The UNIX filename extensions suggest the type of destination libraries into which the source code may be transferred.

Table 4-1 Source Code Filenames

UNIX Filename	MVS Member Name
BEACONN.RDO	BEACONN
BEASNA.RDO	BEASNA
MIRRDPLC.c or MIRRDPLC.cbl	MIRRDPLC
MIRRDTPC.c or MIRRDTPC.cbl	MIRRDTPC
TOUPDPLS.c or TOUPDPLS.cbl	TOUPDPLS
TOUPDTPS.c or TOUPDTPS.cbl	TOUPDTPS

4.1.2.3 Translate CICS/ESA Verbs

This step translates the EXEC CICS verbs into program CALL statements of the form required by the selected source language. The source is read from the SYSIN dataset. The translated source program is written to the SYSPUNCH dataset. The translator listing is written to the SYSPRINT dataset.

Different translator modules are provided for different source languages. There are also language-specific parameters for the translation step. Refer to the *IBM CICS/ESA Application Programming Guide* for additional translation options that might apply to your environment.

The translator modules are installed in the CICS/ESA load datasets. This is indicated in the following examples by the CICSxxx.SDFHLOAD entry, where xxx is the CICS/ESA release number.

- [COBOL Language Translator Example](#)
- [C Language Translator example](#)

4.1.2.3.1 COBOL Language Translator Example

The translator module name in the following listing is DFHECP1\$. The parameter COBOL2 indicates that a source module containing COBOL II verbs is to be translated.

Listing COBOL Language Translator Example

```
//TRN EXEC PGM=DFHECP1$,
// PARM='COBOL2,NOS,CICS',REGION=256K
//STEPLIB DD DSN=CICSXXX.SDFHLOAD,DISP=SHR
//SYSIN DD DSN=YOUR.PDS(pgmname),DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSPUNCH DD DSN=&&SYSCIN,
// DISP=(,PASS),UNIT=SYSDA,
// DCP=BLKSIZE=400,
// SPACE=(400,(400,100))
```

4.1.2.3.2 C Language Translator example

The translator module name in the following listing is DFHEDP1\$. The parameter C indicates that a source module containing C verbs is to be translated.

Listing C Language Translator Example

```
//TRN EXEC PGM=DFHEDP1$,
// PARM='C,NOS,CICS',REGION=256K
//STEPLIB DD DSN=CICSXXX.SDFHLOAD,DISP=SHR
//SYSIN DD DSN=YOUR.PDS(pgmname),DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSPUNCH DD DSN=&&SYSCIN,
// DISP=(,PASS),UNIT=SYSDA,
// DCB=BLKSIZE=400,
// SPACE=(400,(400,100))
```

4.1.2.4 Compile the Translated Source File

The next step is to compile the translated source file `&&SYSCIN`. This step generates the following program modules in preparation for the link-edit step:

- `TOUPDPLS` (CICS server DPL module)
- `TOUPDTPS` (CICS server DTP module using CICS APIs)
- `MIRRDPLC` (CICS client DPL module)
- `MIRRDTPC` (CICS client DTP module using CPI-C verbs)
- [COBOL Compiler Example](#)
- [C Compiler Example](#)

4.1.2.4.1 COBOL Compiler Example

The following listing shows a COBOL compiler example.

Listing COBOL Compiler Example

```
//YOURJOB JOB accounting info,name,MSGLEVEL=1
//STEP1 EXEC PGM=DFHCSDUP,REGION=512K,
//      PARM='CSD(READWRITE),PAGESIZE(60),NOCOMPAT'
//STEPLIB DD DSN=CICSxxx.SDFHLOAD,DISP=SHR
//DFHCSD DD UNIT=SYSDA,DISP=SHR,DSN=CICSxxx.DFHCSD
//SYSPRINT DD SYSOUT=A
//SYSIN DD DSN=YOUR.PDS(BEASNA),DISP=SHR
```

4.1.2.4.2 C Compiler Example

Compile and pre-link your C language source. The Oracle Tuxedo Mainframe Adapter for SNA installation library contains C language compiler examples. The following listing shows a C/MVS compiler example.

In this example, the external storage variables must be re-entrant in the load module for the CICS/ESA environment. Additionally, the C language source contains long variable and function names. The pre-link step must be run to resolve the long names and create re-entrant variables from the compiled object. Perform the pre-link step, use the `RENT` option in the compile program, and use the compile output in the link-edit step.

Listing C/MVS Compiler Example

```
//*-----
//* COMPILE STEP:
//*-----
//COMPILE EXEC PGM=CBC310,REGION=2M,COND=(7,LT,TRN),
//      PARM=('OPT(1),LONGNAME,RENT,SOURCE')
//STEPLIB DD DSN=SYS1.SCEERUN,DISP=SHR
//      DD DSN=SYS1.SCBC3CMP,DISP=SHR
//SYMSGS DD DSN=SYS1.SCBC3MSG(EDCMSGE),DISP=SHR
//SYSIN DD DSN=&&SYSCIN,DISP=SHR **FROM TRN STEP
//SYSLIB DD DSN=SYS1.SCEEH.H,DISP=SHR
//      DD DSN=SYS1.SCEEH.SYS.H,DISP=SHR
```

```

//          DD DSN=CICSxxx.SDFHC370,DISP=SHR**CPIC
//          REQUIRED**
//SYSLIN   DD DSN=&&PLNKSET,UNIT=VIO,
//          DISP=(MOD,PASS),SPACE=(TRK,(3,3)),
//          DCP=(RECFM=FB,LRECL=80,BLKSIZE=&SYSLBLK)
//SYSPRINT DD SYSOUT=*
//SYSOUT   DD SYSOUT=*
//SYSCPRT  DD SYSOUT=*
//SYSUT1   DD UNIT=VIO,SPACE=(32000,(30,30)),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200)
//SYSUT4   DD UNIT=VIO,SPACE=(32000,(30,30)),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200)
//SYSUT5   DD UNIT=VIO,SPACE=(32000,(30,30)),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=12800)
//SYSUT6   DD UNIT=VIO,SPACE=(32000,(30,30)),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=12800)
//SYSUT7   DD UNIT=VIO,SPACE=(32000,(30,30)),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=12800)
//SYSUT8   DD UNIT=VIO,SPACE=(32000,(30,30)),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=12800)
//SYSUT9   DD UNIT=VIO,SPACE=(32000,(30,30)),
//          DCB=(RECFM=FB,LRECL=137,BLKSIZE=882)
//SYSUT10  DD SYSOUT=*
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200)
//SYSUT14  DD UNIT=VIO,SPACE=(32000,(30,30)),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=128007)
//PLKED EXEC PGM=EDCPRLK,COND=( (7.LT,C),(7,LT,TRN)),
//          PARM='MAP'
//STEPLIB  DD DSN=SYS1.SCEERUN,DISP=SHR
//SYSMSGSGS DD DSN=SYS1.SCEEMSGP(EDCPMSGE),DISP+SHR
//SYSLIB   DD DSN=&LE370HLQ..SCEECPP,DISP=SHR
//          DD DUMMY
//SYSIN    DD DSN=&&PLNKSET,DISP=(MOD,PASS)
//SYSMOD   DD DSN=&&LOADSET,DISP=(,PASS),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200)
//SYSOUT   DD SYSOUT=*
//SYSPRINT DD SYSOUT=*

```

4.1.2.5 Create the Executable Object

The next step is to take the compiled source and create the executable object. You should load the resulting object module into the application library that is concatenated with the CICS/ESA region datasets.

In the following COBOL and C program examples, `SYSLIN` is the name of the file containing the compiled source concatenated with other necessary executables, including interfaces for the CICS API verbs, interfaces for CPI-C verbs, and interfaces for SNA resource recovery verbs (sync-level 2).

In the following listing, the module is linked as re-entrant and marked with 31-bit mode address-ability. This action is required for the module `MIRRDTPC` which performs CPI-C and SAA Resource/Recovery requests.

You must change the `NAME` to that of the executable being generated.

Listing COBOL Link-Edit Sample for TOUPDPLS, TOUPDTPS, and MIRRDPLC

```
//LKED      EXEC PGM=IEWL,      * ** LINKAGE EDITOR **
//          PARM=AMODE=31,RENT,
//          REGION=512K
//SYSPRINT  DD SYSOUT=*
//SYSLIN    DD DSN=&&LOADSET,DISP=(OLD,DELETE)
//          DD *
//          INCLUDE SYSLIB(DFHECI)
//          ORDER DFHECI
//          NAME xxxxxxxxx(R)
/*
//SYSLIB    DD DSN=CICSxxx.SDFHLOAD,DISP=SHR
//          DD DSN=CICSxxx.SDFHCOB,DISP=SHR
//          DD DSN=SYS1.SCEELKED,DISP=SHR
//          DD DSN=SYS1.SIGYCOMP,DISP=SHR
//SYSLMOD   DD DSN=application.loadlib,DISP=(SHR,PASS)
//SYSUT1    DD UNIT=VIOD,SPACE=(1024,(50,20))
//
```

Listing COBOL Link-Edit Sample for MIRRDTPC

```
//LKED      EXEC PGM=IEWL,      * ** LINKAGE EDITOR **
//          PARM=AMODE=31,RENT,
//          REGION=512K
//SYSPRINT  DD SYSOUT=*
//SYSLIN    DD DSN=&&LOADSET,DISP=(OLD,DELETE)
//          DD *
//          INCLUDE SYSLIB(DFHECI)
//          INCLUDE SYSLIB(DFHCPLC)
//          INCLUDE SYSLIB(DFHCPLRR)
//          ORDER DFHECI
//          NAME xxxxxxxxx(R)
/*
//SYSLIB    DD DSN=CICSxxx.SDFHLOAD,DISP=SHR
//          DD DSN=CICSxxx.SDFHCOB,DISP=SHR
//          DD DSN=SYS1.SCEELKED,DISP=SHR
//          DD DSN=SYS1.SIGYCOMP,DISP=SHR
//SYSLMOD   DD DSN=application.loadlib(xxxxxxx),DISP=(SHR,PASS)
//SYSUT1    DD UNIT=VIOD,SPACE=(1024,(50,20))
//
```

Listing C Link-Edit Sample

```
//LKED      EXEC PGM=HEWL,REGION=4M,
//          PARM='AMODE=31,RENT,
//          COND=( (7,LT,C), (7,LT,PLKED), (7,LT,TRN) )
//SYSLIB    DD DSN=CICSxxx.SDFHLOAD,DISP=SHR
//          DD DSN=SYS1.SCEELKED,DISP=SHR
//SYSLIN    DD DSN=&&LOADSET,DISP=(OLD,DELETE)
//          DD *INCLUDE SYSLIB(DFHELII)
//          INCLUDE SYSLIB(DFHCPLC)
//          INCLUDE SYSLIB(DFHCPLRR)NAME xxxxxxxxx(R)
/*
```

```
//SYSLOAD DD DSN=application.loadlib(xxxxxxxx),DISP=SHR
//SYSUT1 DD UNIT=VIOD,SPACE=(1024,(50,20))
//SYSPRINT DD SYSOUT=*
```

4.1.2.6 Configure the CICS/ESA Application

Your installed Oracle Tuxedo Mainframe Adapter for SNA software contains two sample files that can be used to configure the CICS application:

- The `BEACONN` file contains the CICS/ESA configuration parameters to the host system. These include connection definitions and session definitions.
- The `BEASNA` file contains the application definitions that enable you to perform the installation verification in the CICS/ESA environment. These definitions are required to run the installation verification. They include the program definitions, transaction definitions, and for the CPI-C example, the partner definition.

Caution:

The `BEACONN` file should only be added to the CICS/ESA System Definition (CSD) file if no definitions currently exist. Check with your system administrator.

One method of adding the files is to use the batch utility program, `DFHCSDUP`. The following listing is an example of the *Job Control Language (JCL)* statements you can use to invoke `DFHCSDUP` as a batch program to add the `BEASNA` file:

Listing JCL Example for Invoking DFHCSDUP

```
//YOURJOB JOB accounting info,name,MSGLEVEL=1
//STEP1 EXEC PGM=DFHCSDUP,REGION=512K,
// PARM='CSD(READWRITE),PAGESIZE(60),NOCOMPAT'
//STEPLIB DD DSN=CICSxxx.SDFHLOAD,DISP=SHR
//DFHCSD DD UNIT=SYSDA,DISP=SHR,DSN=CICSxxx.DFHCSD
//SYSPRINT DD SYSOUT=A
//SYSIN DD DSN=YOUR.PDS(BEASNA),DISP=SHR
```

The definitions in the sample member use an Oracle Tuxedo Mainframe Adapter for SNA *Resource Definition Online (RDO)* Group name. You may want to add these definitions to an existing RDO group, or you might consider adding them to your CICS/ESA start-up list if you plan to use them often. (This automatically installs the group on start-up of the CICS/ESA region.) To add the groups to the start-up list, uncomment the following statements in the sample RDO.

```
ADD GROUP(BEACONN) LIST(**YOURLIST**)
ADD GROUP(BEASNA) LIST(**YOURLIST**)
```

In these arguments, `**YOURLIST**` is the name of your startup list.

To manually install the groups after start-up of the CICS/ESA region, issue the following commands from a CICS/ESA terminal session.

```
CEDA I GROUP (BEACONN)
CEDA I GROUP (BEASNA)
```

- [BEACONN File: Connection definition](#)
- [BEACONN File: Session Definition](#)
- [BEASNA File: Program Definition](#)
- [BEASNA File: Remote Program Definition](#)
- [BEASNA File: Transaction Definition](#)
- [BEASNA File: Partner Definition](#)

4.1.2.6.1 BEACONN File: Connection definition

The BEACONN file includes a sample connection definition.

Note: The sample connection definition achieves the minimum requirements for a connection over which the installation verification can be executed. Do not rely on it to provide optimal performance. Consult the *CICS/ESA Resource Definition Guide* for information about adding options not included in the sample.

The name of the sample connection definition is BEA, which is located under the installation group name BEACONN. It looks like this:

Listing Sample Connection Definition in BEACONN File

```
DEFINE CONNECTION (BEA)      GROUP (BEACONN)
    DE (TUXEDO MAINFRAME ADAPTER FOR SNA EXAMPLE RDO CONNECTION)
    ACCESSMETHOD (VTAM)      PROTOCOL (APPC)
    NETNAME (**LOCALLU**)
    ATTACHSEC (LOCAL)        AUTOCONNECT (NO)
```

****LOCALLU**** must be changed to the LU name of the SNA stack as known by VTAM. The **ATTACHSEC** option indicates the level of attach-time user security required for the connection. **LOCAL** is the simplest security. The authorization of the user is taken to be that of the link itself, relying on the authorization validation provided by the remote security utility. **AUTOCONNECT** indicates when the connection is acquired. **NO** is required. This entry means that CICS does not attempt to bind sessions when the connection is established by the stack.



Note:

The Oracle Tuxedo Mainframe Adapter for SNA software must acquire the connection and negotiate the bind when the Oracle Tuxedo Mainframe Adapter for SNA software starts up.

Install the sample connection definition by putting it on the host in a separate group that does not contain existing connection. For example, use the `CEDA INSTALL` command:

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

4.1.2.6.2 BEACONN File: Session Definition

The `BEACONN` file also includes a sample session definition. When placed on the remote host, it defines the logical links by which the ATMI platform local domain communicates with the remote host.

The name of the sample session definition is `BEATEST`, which is located under the installation group named `BEACONN`. The following listing shows an example of a `BEACONN` file session definition.

Listing Sample Session Definition in BEACONN File

```
DEFINE CONNECTION (BEA)      GROUP (BEACONN)
    DE (TUXEDO MAINFRAME ADAPTER FOR SNA EXAMPLE RDO CONNECTION)
    ACCESSMETHOD (VTAM)      PROTOCOL (APPC)
    NETNAME (**LOCALLU**)
    ATTACHSEC (LOCAL)        AUTOCONNECT (NO)
```

`AUTOCONNECT` indicates how the activation of the session is to be negotiated. `YES` enables the CICS/ESA host to negotiate its own winner sessions when the connection is bound. Remember that the Oracle Tuxedo Mainframe Adapter for SNA software must acquire the connection instead of the CICS/ESA host. However, when the stack acquires the connection, it can only bind the number of sessions identified as its winners. Setting the `AUTOCONNECT` parameter to `YES` causes the host to bind winner sessions immediately when the connection is acquired. Otherwise, the host's outbound clients must wait for winner sessions to bind.

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

The `MAXIMUM` option defines the total number of sessions in the set (`**MAX**`) and the total number of contention winner sessions (`**MIN**`). To verify the installation, the total number of winner sessions must include those for the host and the remote stack. The installation verification process allows both sides to execute as the client. The total number of local contention winner sessions plus remote contention winner sessions must equal the number of sessions. The local number of sessions must equal the remote number of sessions.

4.1.2.6.3 BEASNA File: Program Definition

The `BEASNA` file includes a sample program definition, shown in the following listing. Replace the `LANGUAGE` variable `**LANG**` in the sample with either `C` or `COBOL` to identify the source type you have selected for the sample application.

Listing Sample Program Definitions in BEASNA File

```

DEFINE PROGRAM (MIRRDPLC)    GROUP (BEASNA)
  DE (TUXEDO MAINFRAME ADAPTER FOR SNA EXAMPLE CICS DPL CLIENT
    (MIRROR STRING))
  LANGUAGE (**LANG**)    DATALOCATION (ANY)

DEFINE PROGRAM (MIRRDTPC)    GROUP (BEASNA)
  DE (TUXEDO MAINFRAME ADAPTER FOR SNA EXAMPLE CICS DTP CLIENT
    (MIRROR STRING))
  LANGUAGE (**LANG**)    DATALOCATION (ANY)
DEFINE PROGRAM (TOUPDTPS)    GROUP (BEASNA)
  DE (TUXEDO MAINFRAME ADAPTER FOR SNA EXAMPLE CICS DTP SERVER
    (TOUPPER STRING))
  LANGUAGE (**LANG**)    DATALOCATION (ANY)
DEFINE PROGRAM (TOUPDPLS)    GROUP (BEASNA)
  DE (TUXEDO MAINFRAME ADAPTER FOR SNA EXAMPLE CICS DPL SERVER
    (TOUPPER STRING))
  LANGUAGE (**LANG**)    DATALOCATION (ANY)

```

4.1.2.6.4 BEASNA File: Remote Program Definition

The `BEASNA` file also contains a sample remote program definition, shown in the following listing. The program definition is used by the CICS DPL client to identify the remote system and service for the DPL request. Replace the `REMOTESYSTEM` variable `**CONNECTION ID**` in the sample with the name of the connection for the remote LU.

Listing Sample Remote Program Definition in BEASNA File

```

DEFINE PROGRAM (MIRRDPLS)    GROUP (BEASNA)
  DE (TUXEDO MAINFRAME ADAPTER FOR SNA EXAMPLE DPL REMOTE PROGRAM
    DEFINITION)
  LANGUAGE (C)    DATALOCATION (ANY)
  REMOTESYSTEM (**CONNECTION ID**) REMOTENAME (MIRRDPLS)

```

4.1.2.6.5 BEASNA File: Transaction Definition

The `BEASNA` file also contains a sample transaction definition, shown in the following listing.

Listing Sample Transaction Definitions in BEASNA File

```

DEFINE TRANSACTION (DTPS)    GROUP (BEASNA)
  DE (TUXEDO MAINFRAME ADAPTER FOR SNA EXAMPLE CICS DTP SERVER)
  TASKDATALOC (ANY)    PROGRAM (TOUPDTPS)
DEFINE TRANSACTION (H1PL)    GROUP (BEASNA)
  DE (TUXEDO MAINFRAME ADAPTER FOR SNA EXAMPLE CICS DPL CLIENT -
    SYNCLEVEL 1)
  TASKDATALOC (ANY)    PROGRAM (MIRRDPLC)
DEFINE TRANSACTION (H2PL)    GROUP (BEASNA)
  DE (TUXEDO MAINFRAME ADAPTER FOR SNA EXAMPLE CICS DTP CLIENT -
    SYNCLEVEL 2)
  TASKDATALOC (ANY)    PROGRAM (MIRRDPLC)
DEFINE TRANSACTION (H0TP)    GROUP (BEASNA)
  DE (TUXEDO MAINFRAME ADAPTER FOR SNA EXAMPLE CICS DPL CLIENT -

```

```

        SYNCLEVEL 0)
TASKDATALOC (ANY)    PROGRAM (MIRRDTPC)
DEFINE TRANSACTION (H2TP)  GROUP (BEASNA)
DE (TUXEDO MAINFRAME ADAPTER FOR SNA EXAMPLE CICS DTP CLIENT -
    SYNCLEVEL 2)
TASKDATALOC (ANY)    PROGRAM (MIRRDTPC)

```

4.1.2.6.6 BEASNA File: Partner Definition

The sample CICS/ESA client `MIRRDTPC` contains CPI-C verbs. The partner resource definition contains the CPI-C side information needed to allocate a conversation with the ATMI server and information about the remote LU and transaction program.

As shown in the following listing, the `TPNAME` parameter identifies the transaction program that is invoked in the remote system. In this case, the name correlates to the `RNAME` value in the `DM_LOCAL_SERVICES` section of the `DMCONFIG` file. The `RNAME` there must match the `TPNAME` in the partner definition. (Notice in the sample `DMCONFIG` file that a local service definition `MIRROR` exists. The `RNAME` in that definition matches the `TPNAME` in the sample partner definition.)

The profile resource definition can define conversation attributes, in particular `MODENAME`. In the sample, the `PROFILE` parameter can be replaced with a valid profile resource definition. The default profile name for the parameter is `DFHCICSA`. `DFHCICSA` is a CICS-delivered profile.

Use the `NETNAME` specified in the Connection definition of the remote LU to replace `**NETWORK NAME**`.

Listing Sample Partner Definition in BEASNA File

```

DEFINE PARTNER (MIRRDTPS)    GROUP (BEASNA)
DE (TUXEDO MAINFRAME ADAPTER FOR SNA EXAMPLE CICS DTP CLIENT USING CPIC
    VERBS
TPNAME (MIRRORSERV)        PROFILE (**DFHCICSA**)
NETNAME (**NETWORK NAME**)

```

4.1.2.7 View Connection and Session Status

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

```

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

```

4.2 Running the Sample Application

Now you are ready to validate the Oracle Tuxedo Mainframe Adapter for SNA installation by running the sample application. You must have completed the following prerequisites:

- Install the Oracle Tuxedo Mainframe Adapter software

- Configure the ATMI platform and Oracle Tuxedo Mainframe Adapter for SNA local domains
- Configure the CICS/ESA remote domain
- Initialize the stack processes
- Start the servers

The sample application contains several scenarios. When the client runs as an ATMI client, the server runs as a CICS/ESA host. When the client runs as a CICS/ESA client, the server runs as an ATMI server.

- [Running the Application from an ATMI Client](#)
- [Running the Application from a CICS/ESA Client](#)
- [Running the Application from a CICS/VSE Client](#)

4.2.1 Running the Application from an ATMI Client

In this scenario, the `toupclt` client performs a `tpcall` to the CICS/ESA host server. The server converts the text string you enter from lower-case to upper-case letters. The client may be run in transactional or non-transactional mode. The CICS/ESA server may be run as a DTP or DPL program

For example, enter the following command:

```
toupclt -s 0 -t DTP "hello world"
```

In this command, the arguments and options are defined in the following way:

-s (0|2)

Application Sync-Level.

0 = none (default)

2 = Transactional.

-t (DPL|DTP)

Remote Server Program.

DPL = Distributed Program Link (default)

DTP = Distributed Transaction Program

-h

Help

" "

Lowercase text string of up to 1915 characters.

If the ATMI client successfully executes the command, it displays the text string in upper-case letters, for example:

```
"HELLO WORLD"
```

4.2.2 Running the Application from a CICS/ESA Client

The following sections depict two scenarios for running the application from a CICS/ESA client.

- [CICS/ESA Client with CPI-C](#)
- [CICS/ESA Client with DPL](#)

4.2.2.1 CICS/ESA Client with CPI-C

In this scenario, the CICS/ESA client sends a text string to the Oracle Tuxedo Mainframe Adapter for SNA simple server. The string is re-displayed on the client's screen in reverse order.

Enter either of the following commands:

```
H0TP <string>
```

```
H2TP <string>
```

In these commands, the arguments and options are defined in the following way:

```
H0TP = CPI-C Application Sync-Level 0  
H2TP = CPI-C Application Sync-Level 2  
<string> is a text string up to 1915 characters long.
```

If the CICS/ESA client successfully completes the transaction, it displays the text string in reverse order. For example:

You enter:

```
H0TP HELLO WORLD
```

The system returns:

```
DLROW OLLEH
```

4.2.2.2 CICS/ESA Client with DPL

Two transactions are available to execute the same program for the DPL sample. One is a simple request/response with the required sync-level 1, the other is a transactional request/response with sync-level 2.

Enter either of the following commands:

```
H1PL <string>
```

```
H2PL <string>
```

In these commands, the arguments and options are defined in the following way:

```
H1PL = DPL Application Sync-Level 1
H2PL = DPL Application Sync-Level 2
<string>is a text string up to 955 characters long.
```

If the CICS/ESA client successfully completes the transaction, it displays a reverse (or mirror) image of the text string concatenated to the input text string. For example:

You enter:

```
H1PL HELLO WORLD
```

The system returns:

```
HELLO WORLD : DLROW OLLEH
```

4.2.3 Running the Application from a CICS/VSE Client

The CICS/VSE sample applications are virtually identical to the CICS/ESA sample applications, except the file names have the `.vse` extension. The same scenarios for running the sample applications apply for both client categories. Other noteworthy differences are minor syntax changes in the sample applications to accommodate the CICS/VSE operating system and lack of CICS/VSE support for a CPI-C interface.

- [CICS/VSE Sample Applications](#)
- [Other Considerations](#)

4.2.3.1 CICS/VSE Sample Applications

The following is a list of the CICS/VSE sample applications:

- `BEASNA.RDO.VSE`
- `MIRRDPLC.c.VSE`
- `TOUPDPLS.c.VSE`
- `TOUPDTPS.c.VSE`
- `TOUPDTPS.cb1.VSE`

4.2.3.2 Other Considerations

Some FTP operations may result in mistranslation of C-language special characters. It may be necessary for you to edit the source file(s) and correct the mistranslations of brackets `[]` and parallel bar `| |` characters.

CSMI and CVMI use profile `DFHCICST` with the attribute `INBFMH = NO`. When using DTP transactions, you are required to set `INBFMH = ALL`; otherwise, an `ABEND AXFQ` occurs.

CICS/VSE transactions should be defined with the security attribute `RSLC=NO`; otherwise an `EIBRCODE (NOTAUTH)` is returned to the client.

5

Providing Security

Security refers to techniques for ensuring that data stored in a computer or passed between computers is not compromised. Most security measures involve passwords and data encryption. A password is a secret word or phrase that gives a user access to a particular program or system, and data encryption is the translation of data into a form that is unintelligible without a deciphering mechanism.



Note:

All references to ATMI files, functions, and documentation apply to Tuxedo files, functions, and documentation.

This chapter contains the following topics:

- [Understanding Oracle Tuxedo Mainframe Adapter for SNA Security](#)
- [Mapping User IDs](#)
- [Using Link-Level Encryption](#)
- [Using TLS Encryption](#)
- [Using TCP/IP Link Authentication](#)

5.1 Understanding Oracle Tuxedo Mainframe Adapter for SNA Security

Distributed applications such as those used for electronic commerce (e-commerce) offer many access points for malicious people to intercept data, disrupt operations, or generate fraudulent input. The more distributed a business becomes, the more vulnerable it is to attack. Thus, the distributed computing software or middleware, upon which such applications are built must provide security.

Oracle Tuxedo Mainframe Adapter for SNA works with the ATMI platform to enable the following security capabilities:

- User ID Mapping
- Encryption
- TCP/IP Link Authentication

5.2 Mapping User IDs

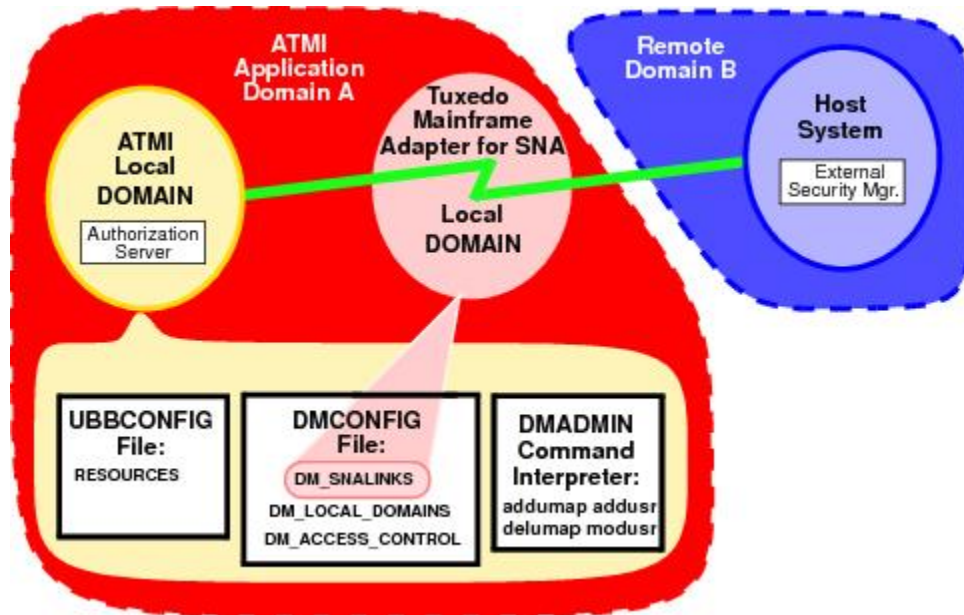
User IDs are used to control access to system resources in the ATMI and mainframe environments. For user IDs to be used by those ATMI and mainframe environments, both sides must have security mechanisms in place. For the ATMI domain, the security mechanism is the Authorization Server. For the host system, the security mechanism is the

External Security Manager. The following figure illustrates Oracle Tuxedo Mainframe Adapter for SNA security elements.

The Oracle Tuxedo Mainframe Adapter for SNA software allows user IDs to be shared between domains in two ways:

- ATMI-to-Host User ID Mapping
- Direct User ID Mapping

Figure 5-1 Oracle Tuxedo Mainframe Adapter for SNA Security Elements



▲ Caution:

Mixed environment security is more complex than depicted in the above figure. A domain without an operational security mechanism in place accepts all transaction requests by treating user IDs as “trusted users.” Refer to the appropriate ATMI product documentation for more detailed information about domain security.

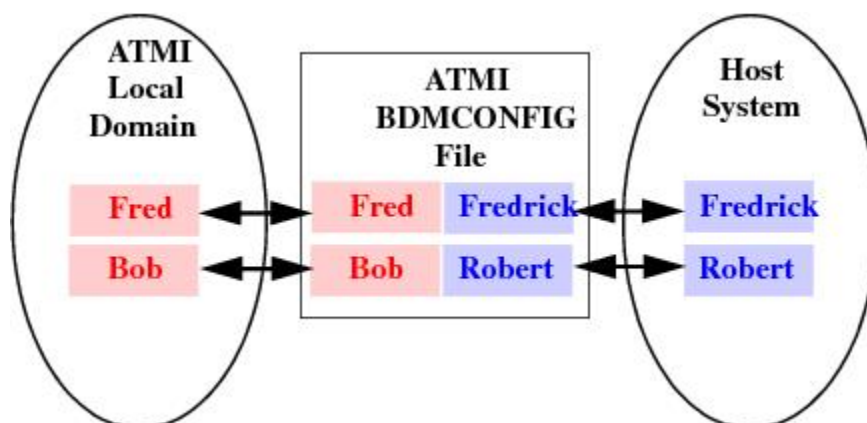
- [ATMI-to-Host User ID Mapping](#)
- [Direct User ID Mapping](#)
- [Configuring User ID Mapping](#)
- [Using dmadmin Commands to Administer User ID Mapping](#)
- [Setting Security Scenario](#)

5.2.1 ATMI-to-Host User ID Mapping

ATMI-to-host user ID mapping associates a user ID in the local domain with a corresponding user ID in the host system. With ATMI-to-host user ID mapping, an ATMI user ID can be different from its counterpart on the host. See the following figure.

The `dmadmin` commands are used to create this kind of mapping. Refer to the [Using dmadmin Commands to Administer User ID Mapping](#) section. These commands change the binary form of the `DMCONFIG` file (called the `BDMCONFIG` file).

Figure 5-2 Typical ATMI-to-host User ID Mapping



5.2.2 Direct User ID Mapping

Direct user ID mapping enables the direct mapping of an ATMI user ID to an identical host user ID. This eliminates the need to use the `dmadmin` commands, as with ATMI-to-host user ID mapping. When this feature is used, any user ID mappings in the `BDMCONFIG` file are bypassed. To enable this feature, specify a command-line parameter with the `GWSNAX` command when starting the Oracle Tuxedo Mainframe Adapter for SNA Gateway. Refer to the [Bypassing User ID Mapping](#) section.



Note:

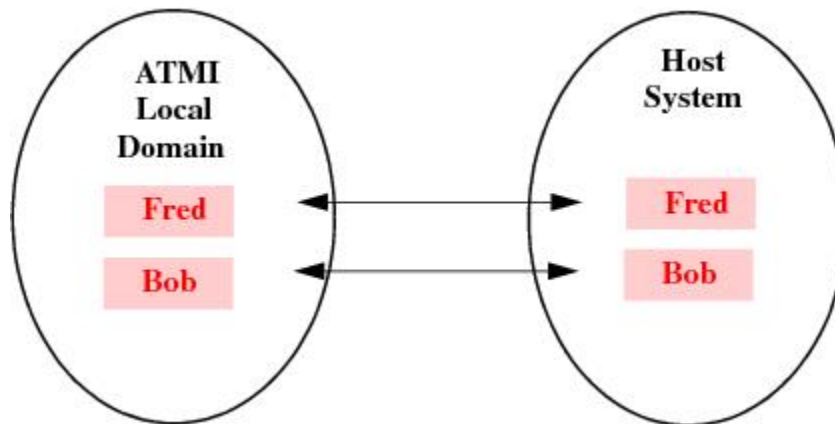
When direct user ID mapping is used, modification or elimination of any `BDMCONFIG` file mapping entries is not necessary.

With direct user ID mapping, the user IDs in the ATMI and host environments must be identical as shown in following figure. When the ATMI local domain initiates a request, the ATMI user ID is applied to the requested host service. When the host initiates a request, the host user ID is applied to the requested ATMI service.

 **Note:**

Identical user IDs must exist in the local domain and in the host domain for direct user ID mapping to be used. With direct mapping, only security level `IDENTIFY` can be supported.

Figure 5-3 Direct User ID Mapping



5.2.3 Configuring User ID Mapping

Specify parameters bearing on local domain and Oracle Tuxedo Mainframe Adapter for SNA security in the `DMCONFIG` and `UBBCONFIG` files in the following four sections:

- `DM_LOCAL_DOMAINS` section of the `DMCONFIG` file
- `DM_SNALINKS` section of the `DMCONFIG` file
- `DM_ACCESS_CONTROL` section of the `DMCONFIG` file
- `RESOURCES` section of the `UBBCONFIG` file
- [Determining Security Parameters](#)
- [Setting DCONFIG File Security Parameters](#)
- [Setting UBBCONFIG File Security Parameters](#)
- [Bypassing User ID Mapping](#)

5.2.3.1 Determining Security Parameters

The combined settings of the `SECURITY` parameters in the `UBBCONFIG` and the `DMCONFIG` files have the following effects:

- When the `DM_LOCAL_DOMAINS` security parameter is set to `NONE` or `APP_PW`, no action is taken by the Gateway with regard to security.
- However, when the `UBBCONFIG` file security parameter is set to `APP_PW`, the application password is validated by an `AUTHSVC` when clients join the application. The `AUTHSVC` is provided by the user application.

If security is to be enforced by both the local domain and the host system for each request inbound from the host system to the local domain, the following settings must be made:

- The UBBCONFIG file SECURITY parameter must be set to one of USER_AUTH, ACL, or MANDATORY_ACL.
- The DMCONFIG file DM_LOCAL_DOMAINS section SECURITY parameter must be set to DM_USER_PW.
- The DMCONFIG file DM_SNALINKS SECURITY parameter must be set to IDENTIFY.
- The SNA stack must be configured with the appropriate parameter for IDENTIFY.
- [Determining Security Parameters for Inbound Requests](#)
- [Determining Security Parameters for Outbound Requests](#)

5.2.3.1.1 Determining Security Parameters for Inbound Requests

Configurations on Oracle Tuxedo Side

The following table shows settings for the SECURITY parameters in the UBBCONFIG and DMCONFIG files required to achieve local domain and host system security combinations for inbound requests from the host system.



Note:

Security setting combinations other than those shown in the tables will have unpredictable results.

Table 5-1 Security Settings for Inbound Requests from Host Systems

Security Combinations		Settings			
Load	Host	UBBCONFIG SECURITY	DM_LOCAL_DOMAINS SECURITY	DM_SNALINKS SECURITY	Remote Verification
No	No	NONE or APP_PW	NONE or APP_PW	LOCAL	Not Applicable
Yes	No	USER_AUTH, ACL, or MANDATORY_ACL	DM_USER_PW	LOCAL	INVALID
No	Yes	NONE or APP_PW	NONE or APP_PW	IDENTIFY	Not Applicable
Yes	Yes	USER_AUTH, ACL, or MANDATORY_ACL	DM_USER_PW	IDENTIFY	UID

For requests sent from the host system, the local domain extracts the remote user ID, or user ID and password, from the conversation start-up request and checks the domain security table. That table contains pairs of local principal user IDs and remote user IDs, maintained on a service-by-service basis. The remote user ID is mapped to the local principal user ID. The

local principal user ID and password are used for further ACL checking, as specified in the `UBBCONFIG` file. If the direct user ID mapping option is specified, the remote user ID is used as the local principal user ID.

When a request is received from the host system, the local domain checks the ACL in the `DMCONFIG` file for the local service to see if requests from the remote domain are permitted. If the `DMCONFIG` file does not contain an ACL for the local service, the service is accessible to all requests.

Configurations on Mainframe Side

Before start, obtain the following information for each VTAM LU 6.2 pair from your VTAM system administrator:

- The LU pair used in TMA and CICS communication: the local LU that can be used by CRM, and the partner LU used by CICS.
- The network ID and the LU identifiers for each member of the pair.
- Whether or not the `NQNAME` option is specified for the ACB. If it is specified, the network-qualified names support is enabled.

You can choose one of the following ways to configure CRM LU in z/OS:

- Define the security profile
 - If the LU pair are enabled to support network-qualified names (`NQN` is specified on the `LUADD` statement), the `RDEFINE` syntax for both LU are:

```
RDEFINE APPCLU local-netid.luid1.remote-netid.luid2 UACC(NONE)
SESSION(SESSKEY(APPCPWD) CONVSEC(ALREADYV))
```

On another side:

```
RDEFINE APPCLU local-netid.luid2.remote-netid.luid1 UACC(NONE) +
SESSION(SESSKEY(APPCPWD) CONVSEC(ALREADYV))
```

- If both LU are not enabled to support network-qualified names (`NONQN` is specified on or used as the default for the `LUADD` statement), and both LU reside on network-id, the `RDEFINE` syntax for them are:

```
RDEFINE APPCLU local-netid.luid1.luid2 UACC(NONE)
SESSION(SESSKEY(APPCPWD) CONVSEC(ALREADYV))
```

On another side:

```
RDEFINE APPCLU local-netid.luid2.luid1 UACC(NONE) +
SESSION(SESSKEY(APPCPWD) CONVSEC(ALREADYV))
```

`local-netid/remote-netid`: The network ID (`NETID`) of the partners. These IDs are specified in the VTAM start option `NETID`, which is in the `ATCSTRxx` member of `SYS1.VTAMLST`.

`luid1/luid2`: The LU names of the partners. In each case, the first LU name specified is the local LU name, and the second LU name is the partner LU name.

 **Note:**

- Do not specify an asterisk (*) or any other generic character for the first two qualifiers (`netid` and `luid`).
- You need to confirm with Mainframe administrator whether `NONQN` or `NQN` must be specified.
- Change VTAM APPL definition of CRM LU. The parameter `SECACPT` is required and its value should be `ALREADYV`.

5.2.3.1.2 Determining Security Parameters for Outbound Requests

If security is to be enforced by both the local domain and the host system for each request outbound from the local domain, the following settings must be configured:

- The `UBBCONFIG` file `SECURITY` parameter must be set to one of `USER_AUTH`, `ACL`, or `MANDATORY_ACL`.
- The `DMCONFIG` file `DM_LOCAL_DOMAINS` section `SECURITY` parameter must be set to `DM_USER_PW`.
- The `DMCONFIG` file `DM_SNALINKS` `SECURITY` parameter must be set to `IDENTIFY` or `VERIFY`.
- The SNA stack must be configured with the appropriate parameter for `IDENTIFY` or `VERIFY`.
- The `ATTACHSEC` level for the connection definition in the host system must be set to `IDENTIFY` or `VERIFY` to match the `DMCONFIG` file `DM_SNALINKS` `SECURITY` parameter.

Configurations on Oracle Tuxedo Side

The following table shows settings for the `SECURITY` parameters in the `UBBCONFIG` and `DMCONFIG` files required to achieve local domain and host system security combinations for outbound requests.

 **Note:**

Security setting combinations other than those shown in the tables will have unpredictable results.

Table 5-2 Security Settings for Outbound Requests from Local Domain

Security Combinations		Settings			
Local	Host	UBBCONFIG SECURITY	DM_LOCAL_DOM AIN S SECURITY	DM_SNALINKS SECURITY	Remote Verification
No	No	NONE or APP_PW	NONE or APP_PW		Not Applicable

Table 5-2 (Cont.) Security Settings for Outbound Requests from Local Domain

Security Combinations		Settings			
Local	Host	UBBCONFIG SECURITY	DM_LOCAL_DOMAINS SECURITY	DM_SNALINKS SECURITY	Remote Verification
Yes	No	USER_AUTH, ACL, or MANDATORY_ACCESS	DM_USER_PW		Not Applicable
No	Yes	NONE or APP_PW	NONE or APP_PW		INVALID
Yes	Yes	USER_AUTH, ACL, or MANDATORY_ACCESS	DM_USER_PW		UID or UID+PW

For a request sent to the host system, the local principal user ID is located in the domain security table and the associated remote user ID, or user ID and password, are put into the conversation start-up request before being sent over the LU6.2 conversation. This situation occurs if `SECURITY` is set to `IDENTIFY` or `VERIFY` in the `DM_SNALINKS` section of the `DMCONFIG` file. If the direct user ID mapping option is specified, the local principal user ID is put into the conversation startup request.

Configurations on Mainframe Side

On Mainframe side, set the following:

1. Set these parameters to `YES` in the CICS system initialization configuration file:

```
SEC=YES
XTRAN=YES
```

When they are specified, only the users defined can access corresponding transactions. You can define valid users in the profile using RACF, for example: `PERMIT * CLASS(TCICSTRN) ID(GUMENG) ACCESS(READ) *` can be replaced by the transaction name if you want to control individual transaction.

2. Configure the SNA stack with the appropriate parameter for `IDENTIFY` or `VERIFY`.
3. Set the `ATTACHSEC` level for the connection definition in the host system to `IDENTIFY` or `VERIFY` to match the `DMCONFIG` file `DM_SNALINKS SECURITY` parameter.

5.2.3.2 Setting DMCONFIG File Security Parameters

Three sections in the `DMCONFIG` file contain parameters affecting Oracle Tuxedo Mainframe Adapter for SNA control of access to the ATMI local domain:

- `DM_LOCAL_DOMAINS` section contains a `SECURITY` parameter which specifies the type of security enforced for the ATMI local domain.
- `DM_SNALINKS` section contains a `SECURITY` parameter that records the security in effect for the host system.

- `DM_ACCESS_CONTROL` section contains local access control lists used by the ATMI local domain to associate local resources with host systems permitted to have access to them.

▲ Caution:

Do not delete the `DMCONFIG` binary file before running the `dmloadcf` command. Tables of remote users, remote passwords, and remote mappings are stored in this file. If deleted, all security information must be re-entered.

- [DM_LOCAL_DOMAINS Section](#)
- [DM_SNALINKS Section](#)
- [DM_ACCESS_CONTROL Section](#)

5.2.3.2.1 DM_LOCAL_DOMAINS Section

The `SECURITY` parameter settings in this section work in conjunction with the `SECURITY` parameter in the `RESOURCES` section of the ATMI local domain's `UBBCONFIG` file to establish how Oracle Tuxedo Mainframe Adapter for SNA controls access to the ATMI local domain. The parameter takes the form:

SECURITY = {value}

In this parameter, `value` can be set as:

NONE

No security is enforced.

APP_PW

No security is enforced.

DM_USER_PW

User and password security is enforced.

If this parameter is set to `NONE` or `APP_PW`, the local domain takes no action with regard to security. If this parameter is set to `DM_USR_PW`, the local domain enforces security according to the setting in the `UBBCONFIG` file (refer to “Setting `DMCONFIG` File Security Parameters”).

5.2.3.2.2 DM_SNALINKS Section

This section of the `DMCONFIG` file is dedicated to Oracle Tuxedo Mainframe Adapter for SNA parameters. It records the security in effect for the host system. It correlates to the values set for the `ATTACHSEC` parameter in the connection resource definition. In the following parameter definition, *remote* refers to the ATMI domain and *local* refers to the host system. The parameter takes the form:

SECURITY_TYPE = {value}

In this parameter, `value` can be set as:

LOCAL

Specifies that a user identifier is not to be supplied by the remote system. `LOCAL` is the default value.

IDENTIFY

Specifies that a user identifier is expected on every attach request. All remote users of a system must be identified to the remote external security manager.

VERIFY

Attaches a user ID and valid password to the remote region. The user ID and password are controlled by the region's external security manager.

PERSISTENT

Not fully supported. Functions the same as `VERIFY`.

MIXIDPE

Not fully supported. Functions the same as `VERIFY`.

The values `LOCAL` and `IDENTIFY` are roughly equivalent to `NONE` and `APP_PW` for the `SECURITY` parameter in the `DMCONFIG` file.

5.2.3.2.3 DM_ACCESS_CONTROL Section

This section contains local ACL used by the ATMI local domain to restrict access by remote regions to local resources. Refer to the "ATMI Security Administration" section in the *Oracle Tuxedo 22c* documentation.) Each entry consists of an `ACL_NAME` resource identifier along with a list of required parameters designating remote domains permitted to access the resource. If no entry exists for a local service, the service is accessible to all remote domains.

5.2.3.3 Setting UBBCONFIG File Security Parameters

The `RESOURCES` section in this file contains a `SECURITY` parameter that works in conjunction with the `SECURITY` parameter in the `DMCONFIG` file to establish how Oracle Tuxedo Mainframe Adapter for SNA controls access to the ATMI local domain. This parameter takes the form:

```
SECURITY = {value}
```

In this parameter, `value` can be set as:

NONE

No security is enforced (default).

APP_PW

Requires password authorization for the Gateway and administrative tools to connect to the local application.

USER_AUTH

Same as `APP_PW`, but additional authorization is required on a per-user basis.

ACL

Same as `USER_AUTH`, but additional access-control checks are done on service names, queue names, and event names. If no *Access Control Lists (ACL)* exists for a given name, access is granted.

MANDATORY_ACL

Same as `ACL`, but if no ACL exists for a given name, access is denied.

In most cases, the `UBBCONFIG` file has already been configured and you do not need to establish the `SECURITY` parameter settings, but examining this file enables you to see how Oracle Tuxedo Mainframe Adapter for SNA enforces security.

If this parameter is set to `NONE`, no security is enforced. If set to `APP_PW`, the local ATMI domain's Authorization Server prompts for the application password. If set to `USER_AUTH`, `ACL`, or `MANDATORY_ACL`, the qualified security is enforced as specified.

5.2.3.4 Bypassing User ID Mapping

To use direct user ID mapping, use the `-m` parameter in the `GWSNAX` process start-up command line entry. This parameter allows you to establish direct user ID mapping, rather than ATMI-to-host user ID mapping.

 **Note:**

If you bypass user ID mapping, the local and host domains must have identical user IDs in effect, otherwise a security error occurs.

For example, to set the Gateway server process to bypass user ID mapping, enter a command in the following format:

```
GWSNAX SRVGRP = <groupname> SRVID = <number> CLOPT = "-A -- -m"
```

Refer to `GWSNAX` in Appendix A, [Administration Mode Commands](#) for more information.

5.2.4 Using `dmadmin` Commands to Administer User ID Mapping

When ATMI-to-host user ID mapping is used, you must create mappings in the `BDMCONFIG` file.

 **Note:**

If the direct user ID mapping option is specified, creation of mappings in the `BDMCONFIG` file is not necessary. Any mappings in the `BDMCONFIG` file are ignored.

User ID mapping between the local domain and the host system is configured using the `addumap`, `addusr`, `delumap`, `modusr`, and `delusr` commands of the `dmadmin` utility to accomplish the following tasks:

- Adding a User ID and Password
- Mapping a User ID
- Removing User ID Mapping
- Deleting a User ID and Password
- Modifying a Password

Refer to Appendix A, [Administration Mode Commands](#) for more information about each ATMI command.

To use these commands, enter `dmadmin` at the system prompt. At the `dmadmin ">"` prompt, enter the commands as described.

- [Adding a User ID and Password](#)
- [Mapping a User ID](#)
- [Removing User ID Mapping](#)
- [Deleting a User ID and Password](#)
- [Modifying a Password](#)

5.2.4.1 Adding a User ID and Password

Use the `addusr` ATMI command to add an ATMI local domain user ID and password to the remote domain user and password file. Enter the following command:

```
addusr {-d} local_domain_id {-R} remote_domain_id {-u} remote_userid  
[{-w}]
```

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

-d

Specifies the name of the local domain with which the user ID and password are associated.

-R

Specifies the name of the remote domain to which the user ID and password are added.

-u

Specifies the user name to be added. Enter the user's password when prompted.

-w

Specifies not to prompt for password. Use when running with `IDENTIFY`.

5.2.4.2 Mapping a User ID

Use the `addumap` ATMI command to map a local domain principal user ID number to a remote domain user name. The user ID must be added before it can be mapped. Refer to the [Adding a User ID and Password](#) section. Enter the following command:

```
addumap {-d} local_domain_id {-R} remote_domain_id  
{-p} local_principal_userid {-u} remote_userid
```

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

-d

Specifies the name of the local domain with which the user ID is associated.

-R

Specifies the name of the remote domain to which the user ID is mapped.

-p
Specifies the local principal user ID number defined in the `tpusr`.

-u
Specifies the remote user name as defined in the security application of the remote domain.

5.2.4.3 Removing User ID Mapping

Use the `delumap` ATMI command to remove the mapping for a local domain principal user ID to a remote domain user name. Enter the following command:

```
delumap {-d} local_domain_id {-R} remote_domain_id  
{-p} local_principal_userid {-u} remote_userid
```

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

-d
Specifies the name of the local domain with which the user ID is associated.

-R
Specifies the name of the remote domain to which the user ID is mapped.

-p
Specifies the local principal user ID number defined in the `tpusr`.

-u
Specifies the remote user name as defined in the security application of the remote domain.

5.2.4.4 Deleting a User ID and Password

Use the `delusr` ATMI command to remove a local ATMI domain user ID and password from the remote domain user and password file. The mapping for a user ID must be removed before the user ID can be removed. Enter the following command:

```
delusr {-d} local_domain_id {-R} remote_domain_id {-u} remote_userid
```

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

-d
Specifies the name of the local domain with which the user ID and password are associated.

-R
Specifies the name of the remote domain from which the user ID and password are to be deleted.

-u
Specifies the user name to be deleted.

5.2.4.5 Modifying a Password

Use the `modusr` ATMI command to modify a local domain user's password recorded in a remote domain's user and password file. Enter the following command:

```
modusr {-d} local_domain_id {-R} remote_domain_id {-u} remote_userid  
[{-w}]
```

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

-d

Specifies the name of the local domain the user ID and password are associated with.

-R

Specifies the name of the remote domain in which the user ID and password are to be modified.

-u

Specifies the user name to be modified. Enter the user's password when prompted.

-w

Specifies not to prompt for password. Use when running with `IDENTIFY`.

5.2.5 Setting Security Scenario

This section provides an example of step-by-step instructions for setting up security in an application that has already been configured.

- [Configuring Security in the ATMI Domain](#)
- [Configuring Security in the Local Domain](#)
- [Configuring Security in the Remote Domain](#)
- [Setting the Security Level to IDENTIFY](#)

5.2.5.1 Configuring Security in the ATMI Domain

1. Edit the `UBBCONFIG` file.
 - a. In the `RESOURCES` section, add `SECURITY USER_AUTH`.
 - b. In the `SERVERS` section, add the `AUTHSVR` server.

 **Note:**

`SECURITY USER_AUTH` level implies that application passwords, user IDs, and user passwords are required to join the application. `AUTHSVR` is the ATMI-supplied authentication server. It advertises the service `AUTHSVC`.

2. Enter the `tmloadcf` command to load the ATMI configuration, for example:

```
tmloadcf -y ubbconfig.sna
```

3. Set the application password. (The `tmloadcf` command prompts for the application password.)
4. Add users to the ATMI domain by using the `tpusradd` command. The command prompts for each password, for example: `tpusradd me`
(Enter password for me.)

 **Note:**

Do not use the command `tpaddusr`.

5. Modify the ATMI client to specify security parameters in the `tpinit` call. The following listing 4-1 is an example of the code to do this.

Listing Security Parameters Added to `tpinit` Call

```

    TPINIT *tpinitbuf;
    char passwd[30];
    int security_level;
/* Initialize security parameters */
    if ((tpinitbuf = (TPINIT *) tmalloc("TPINIT", NULL,
    TPINITNEED(sizeof(passwd)))) == NULL)
    {
        userlog("tpalloc tpinit failed %s \n", tpstrerror(tperrno));
        exit(1);
    }
    strcpy(tpinitbuf->usrname, "");
    strcpy(tpinitbuf->cltname, "");
    strcpy(tpinitbuf->passwd, "");
    strcpy(tpinitbuf->grpname, "");

/* Determine level of enforced security */
    security_level = tpchkauth();

    if ((security_level == TPSYSAUTH) || (security_level ==
    TPAPPAUTH))
    {
        fprintf(stdout, "\nApplication passwd required.");
        fprintf(stdout, "\nApplication passwd:");
        gets(tpinitbuf->passwd);
    }
if (security_level == TPAPPAUTH)
{
    fprintf(stdout, "\nUser Name required.");
    fprintf(stdout, "\nUser Name:");
    gets(tpinitbuf->usrname);
    fprintf(stdout, "\nUser Password required.");
    fprintf(stdout, "\nUser Password:");
    gets(passwd);
    strcpy(&tpinitbuf->data, passwd);
    tpinitbuf->datalen=strlen(passwd);
}
if (tpinit(tpinitbuf) == -1)
{
    userlog("TPINIT %s \n", tpstrerror(tperrno));

```

```
        exit(1);
    }
```

6. Verify security in the ATMI domain by running the client.
7. Enter the `dmloadcf` command to load the domain configuration. For example:

```
dmloadcf -y dmconfig.sna
```

8. Enter the `tmboot` command to boot the ATMI domain, for example: `tmboot -y`
9. Configure security for the SNA domain by editing the `DMCONFIG` file.
 - a. In the `DM_LOCAL_DOMAINS` section, add the parameter: `SECURITY=DM_USER_PW`
 - b. In the `DM_SNALINKS` section, add the parameter for the remote link:

```
SECURITY=VERIFY
```
10. Add the user name mapping for the remote domain by invoking `dmadmin` and using the `addumap` command to map local user IDs to remote user IDs. For example:

```
dmadmin
>addumap -d myldom -R myrdom -p localme -u REMOTEME
```

11. Add a password for remote user IDs for the remote domain by invoking `dmadmin` and using the `addusr` command to provide remote password(s). For example:

```
dmadmin
>addusr -d myldom -R myrdom -u REMOTEME
```

(The system responds with the following prompts:

```
ERROR: Enter Remote User's Password:
ERROR: Re-enter Remote User's Password:)
```

5.2.5.2 Configuring Security in the Local Domain

To configure security in the local domain:

- Set the security parameters in the SNA stack.

Refer to the appropriate stack documentation.

5.2.5.3 Configuring Security in the Remote Domain

Change the security of connection definitions on the mainframe host by doing the following:

1. Expand the group that contains the connection definitions. For example: `CEDA EX GR (MYCONNGRP)`
Replace `MYCONNGRP` with the name of the group that contains your connection definitions.
2. Alter security of each connection definition by changing the value of `ATTACHSEC` to `VERIFY` on each connection definition.

3. Put each connection out of service by inquiring on the connection's `CEMT I CONN(MYCN)` and tabbing to the connection, then changing the `INS` entry to `OUT`.
4. Install the modified connection definitions. For example: `CEDA I GR(MYCONNGRP)`
Replace `MYCONNGRP` with the name of the group that contains your connection definitions.
5. The security definition is complete. Run the application.

5.2.5.4 Setting the Security Level to IDENTIFY

To configure the previous example with a security level of `IDENTIFY`, complete the following steps:

1. Change the `SECURITY` parameter in the `DMCONFIG` file to `IDENTIFY`.
2. Change the `ATTACHSEC` parameter on the connection to `IDENTIFY`.

5.3 Using Link-Level Encryption

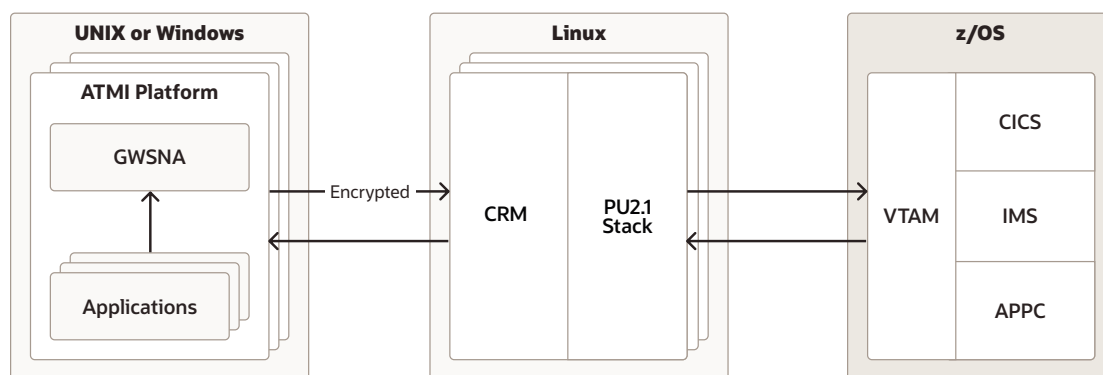
To establish secure communications between the Communication Resource Manager (CRM) and the Gateway (GWSNAX) over a distributed network, Oracle Tuxedo Mainframe Adapter for SNA uses a link-level encryption process.

- [Illustration of Encryption Process](#)
- [Configuring the Oracle Tuxedo Mainframe Adapter for SNA Gateway and CRM](#)

5.3.1 Illustration of Encryption Process

As illustrated in the following diagram, this encryption feature only applies to the link between the Oracle Tuxedo Mainframe Adapter for SNA Gateway and the CRM.

Figure 5-4 Encrypted Links



The encryption process occurs in the following way:

1. When the Gateway establishes a connection to the CRM, the entities exchange messages to determine if encryption is enabled.
2. If both entities have encryption capability, a negotiation is performed to determine the level of encryption established.
Each process has a range of acceptable encryption levels, as specified on the process start-up command line. The lowest common level of encryption is used.

 **Note:**

When encryption is established for communications between the CRM and the Gateway, system performance may deteriorate. The higher the encryption level, the more likely deterioration may occur.

5.3.2 Configuring the Oracle Tuxedo Mainframe Adapter for SNA Gateway and CRM

To configure the Oracle Tuxedo Mainframe Adapter for SNA Gateway:

1. Determine acceptable range of encryption levels (`min` and `max`).
2. Edit the `GWSNAX` entry in the `UBBCONFIG` file to add the `-n` option with the desired `min` and `max`.
See `GWSNAX` in Appendix A, [Administration Mode Commands](#).

To configure the CRM:

1. Determine acceptable range of encryption levels (`min` and `max`).
2. Configure the CRM in one of the following ways:
 - If the CRM is started from the command line, add the `-n` option with the desired `min` and `max`, as described in `SNACRM` in Appendix A, [Administration Mode Commands](#).
 - If the CRM is started as an ATMI server, modify the `SNACRM` server entry to contain the `-n` option with the desired `min` and `max`, as described in `SNACRM` in Appendix A, [Administration Mode Commands](#).

If `crmlkoff`, `crmlkon`, or `crmdown` are used with encrypted CRM, no additional command line arguments are required.

5.4 Using TLS Encryption

To establish TLS between the Communication Resource Manager (CRM) and the Gateway (GWSNAX) over a distributed network, you require to enable TLS on both GWSNAX and CRM side in Oracle Tuxedo Mainframe Adapter for SNA.

The encryptions process occurs in the following way:

1. When the Gateway establishes a connection to the CRM, the entities exchange messages to determine if TLS encryption is enabled.
2. If both entities have TLS encryption capability, a negotiation is performed to determine the cipher to be used.

Each process has a range of acceptable cipher suites, as specified on the process start-up command line. The strongest supported cipher suite is used.

- [Configuring the Oracle Tuxedo Mainframe Adapter for SNA Gateway and CRM](#)
- [Security Enforcement](#)
- [Samples of TLS Encryption Configuration File for CRM](#)

5.4.1 Configuring the Oracle Tuxedo Mainframe Adapter for SNA Gateway and CRM

To configure the Oracle Tuxedo Mainframe Adapter for SNA Gateway:

1. Edit the `GWSNAX` entry in the `UBBCONFIG` file to add the `-n` option in `CLOPT` with the desired `min` and `max` to determine acceptable range of cipher suites.
2. Define the `SEC_PRINCIPAL_NAME`, `SEC_PRINCIPAL_LOCATION`, and `SEC_PRINCIPAL_PASSVAR` parameters for the `GWSNAX` process in `*MACHINES` sections in the `UBBCONFIG` file.

To configure the CRM deployed on Linux:

1. Define the CRM `-n` option with the desired `min` and `max` to determine acceptable range of cipher suites (`-n SSL:min:max`) in CRM command line.
2. Define `-s` to specify TLS configuration file.
3. Edit the `SEC_PRINCIPAL_NAME`, `SEC_PRINCIPAL_LOCATION`, and `SEC_PRINCIPAL_PASSVAR` parameters for CRM process in the TLS configuration file specified by the `-s` option.

To configure the CRM deployed on z/OS:

1. Define `-s` to specify TLS configuration file in CRM command line.
2. Edit `GSK_KEYRING_FILE`, `GSK_KEYRING_PW`, and `GSK_KEYRING_LABEL` in the TLS configuration file specified by the `-s` option.
`GSK_KEYRING_FILE` indicates the absolute certificate file path for the secure session or TLS environment. `GSK_KEYRING_PW` points to the password for the certificate store file to be used for the secure session or TLS environment. `GSK_KEYRING_LABEL` specifies the certificate label associated with the certificate in the certificate store to be used for the secure session or TLS environment.

5.4.2 Security Enforcement

TMA SNA 22c requires TLS 1.2 by default for a secure configuration. `GWSNAX` fails to start if `CLOPT '-n SSL'` is not set, and also, CRM fails to start if `CLOPT '-n SSL'` is not set. To interoperate with older Tuxedo versions, customers may use the environment variable `TM_TLS_FORCE_VER` when accepting TLS 1.0 or 1.1 connections.

Default cipher suites supported include the following:

```

TLS_RSA_WITH_AES_256_CBC_SHA256
TLS_RSA_WITH_AES_256_GCM_SHA384
TLS_RSA_WITH_AES_128_CBC_SHA256
TLS_RSA_WITH_AES_128_GCM_SHA256
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256

```

Elliptic Curve Cryptography (ECC) based cipher suites, such as

```

TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 and
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256, must also be supported by ECC-based TLS
certificates, and the environment variable TM_MIN_PUB_KEY_LENGTH must be set to 0.

```

If customers prefer to use other cipher suites, for example, to interoperate with older Tuxedo versions, the `TM_CIPHERSUITES` environment variable can be used.

RSA requires a minimum key length of 2048. The TMA SNA gateway determines the key length when loading the key/certificate, and fails the load if the key length is less than 2048. `TM_MIN_PUB_KEY_LENGTH` is an environment variable that specifies the minimum key length that customers may use.

Set the environment variable `TUX_SSL_HOSTNAME_VALIDATE` to `Y` to enable TLS hostname validation.

The following environment variable provides backward compatibility for users who prefer to maintain the old behavior as in TMA SNA 12.2.2. If the following variable `TM_ALLOW_NOTLS` is set to `Y`, it allows non-TLS connections.

5.4.3 Samples of TLS Encryption Configuration File for CRM

Configuration file for CRM deployed on Linux:

```
SEC_PRINCIPAL_NAME=mysample
SEC_PRINCIPAL_LOCATION=/storage/cert
SEC_PRINCIPAL_PASSVAR=password
SEC_TRACE_LEVEL=3
```

Configurations file for CRM deployed on z/OS:

```
GSK_KEYRING_LABEL=mysample
GSK_KEYRING_FILE=/storage/CERTDB
GSK_KEYRING_PW=password
GSK_TRACE_LEV=3
```

5.5 Using TCP/IP Link Authentication

In addition to encryption, Oracle Tuxedo Mainframe Adapter for SNA uses an authentication process to establish secure communications between the CRM and Gateway over a distributed network.

The following table lists the processes that support authentication.

Table 5-3 CRM Processes Supporting Authentication

Process	Encryption Capability
<code>crmlkon</code>	Supports one-way authentication; CRM authenticates the <code>crmlkon</code> program, not vice versa
<code>crmlkoff</code>	Supports one-way authentication; CRM authenticates the <code>crmlkoff</code> program, not vice versa
<code>crmdown</code>	Supports one-way authentication; CRM authenticates the <code>crmdown</code> program, not vice versa
<code>GWSNAX</code>	Supports two-way authentication
<code>CRM</code>	Supports two-way authentication

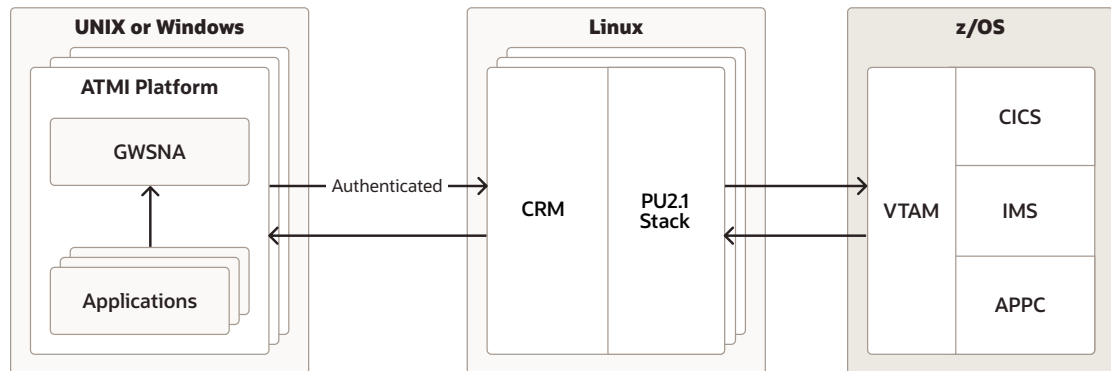
- [Illustration of Authentication Process](#)

- [Configuring the Oracle Tuxedo Mainframe Adapter for SNA Gateway and CRM for Authentication](#)

5.5.1 Illustration of Authentication Process

As illustrated in the following diagram, this authentication feature only applies to the link between the Gateway and the CRM.

Figure 5-5 Authenticated Links



When the Gateway establishes a connection to the CRM, the following events occur:

1. Each entity issues a challenge.
 - The challenge is based on a random number combined with an authentication key.
 - The authentication key is contained in a key file designated by the process command line specification.
2. Each entity issues a response to the challenge it receives. This response is based on the challenge combined with the entity's authentication key.
3. Each entity verifies the response by comparing the response to its own calculated result.
 - If the challenge/response exchange fails, the connection is closed and an error is logged.
 - If the challenge/response succeeds, full communications are enabled.

5.5.2 Configuring the Oracle Tuxedo Mainframe Adapter for SNA Gateway and CRM for Authentication

To configure the Gateway for authentication, complete the following steps:

1. Establish an authentication key file.
 - Create a text file containing the authentication key. This key should be no more than eight characters. Communicating processes must have the same entry in their key files for authentication to be successful.
 - Store the `keyfile` in a protected location.

2. Use a general command line entry with the following format to establish authentication, as described in `GWSNAX` in Appendix A, [Administration Mode Commands](#):

```
[-u <keyfile>]
```

To configure the CRM:

1. Establish an authentication key file.
 - Create a text file containing the authentication key. This key should be no more than eight characters. Communicating processes must have the same entry in their key files for authentication to be successful.
 - Store the `keyfile` in a protected location.
2. Configure the CRM in one of the following ways:
 - If the CRM is started from the command line, add the `-u<keyfile>` option, as described in `SNACRM` in Appendix A, [Administration Mode Commands](#)
3. If `crmlkoff`, `crmlkon`, or `crmdown` are used with a CRM with authentication enabled, use the `-u<keyfile>` command line option as described in `SNACRM` in Appendix A, [Administration Mode Commands](#).

6

Setting Up Data Translations

Due to differences in platforms, operating systems, and programming languages, the data format used by applications in the ATMI domain differs significantly from that expected by applications in the remote domain. In the ATMI domain, applications are typically written in the C language, using C structures and ATMI typed buffers, such as VIEW and FML. In the remote domain, the C and COBOL programming languages are commonly used, with C structures or COBOL copybook definitions. The translation between these data types for both inbound and outbound messages occurs in the Oracle Tuxedo Mainframe Adapter for SNA Gateway. The Gateway has a number of configuration options for dealing with data conversion between domain types.



Note:

All references to ATMI files, functions, and documentation apply to Tuxedo files, functions, and documentation.

This chapter contains the following topics:

- [Data Conversion](#)
- [Translation Rules for VIEW Data Types](#)
- [Translation Rules for RECORD Data Types](#)
- [Translation Rules for Strings](#)
- [Multi-byte Character Set \(MBCS\) Support](#)
- [ART for CICS Integration Consideration](#)

6.1 Data Conversion

Like other domain gateways, the Oracle Tuxedo Mainframe Adapter for SNA Gateway uses ATMI *typed buffers* to transmit and receive data. Since the remote host application does not understand the typed buffer, the ATMI application must communicate with the host application by using an aggregate data type known as a *record*. A *record* is a flat data area defined by a template that describes the data type and length of each field in the record.

The application developer should determine the format and content of the record structure used on the remote host, as well as the ATMI typed buffer to be used, before configuring the Gateway.

In most cases the conversion between ATMI typed buffers and record formats is handled by the Oracle Tuxedo Mainframe Adapter for SNA Gateway. The service definitions in the `DM_LOCAL_SERVICES` and the `DM_REMOTE_SERVICES` section of the `DMCONFIG` file provide parameters to describe the typed buffer/record combination required for successful communications between the applications.

 **Note:**

The current size of remote host messages is limited to approximately 32K bytes. Any conversions resulting in records larger than 32756 bytes are not supported.

- [Conversion of ATMI Typed Buffers to Records](#)
- [Conversion of Records to ATMI Typed Buffers](#)
- [Data Conversion For DPL Services](#)

6.1.1 Conversion of ATMI Typed Buffers to Records

When an ATMI application sends a typed buffer to a remote host application, the buffer must be converted to a record by the Oracle Tuxedo Mainframe Adapter for SNA Gateway before it is passed to the remote host application. The Gateway uses the service definition to determine what, if any, conversion must be applied to the buffer. The service definition uses the `INBUFTYPE` in both the `DM_LOCAL_SERVICES` and `DM_REMOTE_SERVICES` section of the `DMCONFIG` file to describe the desired conversion.

`INBUFTYPE` is specified in the following way:

```
INBUFTYPE = type:subtype
```

In this parameter definition, `type` must be one of the designated ATMI typed buffers described in the following subsections.

The `subtype` value names a view and is required for certain ATMI typed buffers.

Only one `type:subtype` may be entered for the `INBUFTYPE` parameter.

The following sections describe the conversions performed for each ATMI typed buffer.

- [Data Conversion for STRING Typed Buffer](#)
- [Data Conversion for XML Typed Buffer](#)
- [Data Conversion for X_OCTET/CARRAY Typed Buffers](#)
- [Data Conversion for VIEW/VIEW32/X_C_TYPE/X_COMMON Typed Buffers](#)
- [Data Conversion for FML/FML32 Typed Buffers](#)
- [Data Conversion for RECORD Typed Buffers](#)

6.1.1.1 Data Conversion for STRING Typed Buffer

By default, a null-terminated string is converted to EBCDIC. The null character is part of the converted record. See the [Translation Rules for Strings](#) section for information about using strings in View and FML data types.

6.1.1.2 Data Conversion for XML Typed Buffer

The XML typed buffer is converted to EBCDIC for the length of the buffer.

6.1.1.3 Data Conversion for X_OCTET/CARRAY Typed Buffers

No data conversion is performed on these typed buffers. The ATMI application or remote host application performs all conversion of data fields in the record, including all numeric and EBCDIC conversions.

These typed buffers are used when a data record cannot be described or converted using one of the other *strong* typed buffers. *Strong* means that Oracle Tuxedo Mainframe Adapter for SNA Gateway can understand all data fields and perform the required data conversions.

These typed buffers are also options when the remote service expects many styles of data aggregation (multiple record types), because the `INBUFTYPE` parameter is limited to one `type:subtype`.

6.1.1.4 Data Conversion for VIEW/VIEW32/X_C_TYPE/X_COMMON Typed Buffers

A subtype is required for these typed buffers. The subtype is the name of the view that describes the remote host record. The ATMI buffer is converted from a C structure to the record, including EBCDIC conversion, using the compiled VIEW file. By default, the record is a COBOL structure, mapped by the remote host application using a COBOL copybook. See [Translation Rules for VIEW Data Types](#) for more conversion options.

6.1.1.5 Data Conversion for FML/FML32 Typed Buffers

A subtype is required for these typed buffers. The subtype is the name of the view that describes the remote host record. The data in the typed buffer is *Field Manipulation Language (FML)* data. The Oracle Tuxedo Mainframe Adapter for SNA Gateway converts the buffer to a record described by the view, including EBCDIC conversion.

The ATMI buffer is converted from an FML typed buffer to a C structure using the subtype compiled VIEW file. The C structure is then converted to the record using the same subtype compiled VIEW file. By default, the record is a COBOL structure that is mapped by the remote host application using a COBOL copybook.

6.1.1.6 Data Conversion for RECORD Typed Buffers

A subtype is required for this typed buffer. The subtype is the name of the RECORD that describes the remote host record. The ATMI buffer is converted from a RECORD typed buffer to the record, including EBCDIC conversion, using the compiled RECORD file. By default, the record is a COBOL structure, mapped by the remote host application using a COBOL copybook.

6.1.2 Conversion of Records to ATMI Typed Buffers

When a remote application sends a record to an ATMI application, the record must be converted to an ATMI typed buffer by the Oracle Tuxedo Mainframe Adapter for SNA Gateway before it is passed to the ATMI application. The Gateway uses the service definition to determine what, if any, conversion must be applied to the host record. The service definition uses the `OUTBUFTYPE` in both the `DM_LOCAL_SERVICES` and `DM_REMOTE_SERVICES` section of the `DMCONFIG` file to describe the desired conversion.

OUTBUFTYPE is specified in the following way:

```
OUTBUFTYPE=type:subtype
```

In this parameter definition, `type` must be one of the designated ATMI typed buffers described in the following subsections. The `type` not only determines the typed buffer, but also describes the host record.

The `subtype` value names a view and is required for certain ATMI typed buffers.

Only one `type:subtype` may be entered for the `OUTBUFTYPE` parameter.

The following sections describe the conversions performed for each ATMI typed buffer.

- [Data Conversion for STRING Typed Buffer](#)
- [Data Conversion for XML Typed Buffer](#)
- [Data Conversion for X_OCTET/CARRAY Typed Buffers](#)
- [Data Conversion for VIEW/VIEW32/X_C_TYPE/X_COMMON Typed Buffers](#)
- [Data Conversion for FML/FML32 Typed Buffers](#)
- [Data Conversion for RECORD Typed Buffers](#)

6.1.2.1 Data Conversion for STRING Typed Buffer

The null terminated string is converted to ASCII. The converted string is moved to an ATMI STRING typed buffer. See the [Translation Rules for Strings](#) section for information about using strings in View and FML data types.

6.1.2.2 Data Conversion for XML Typed Buffer

The XML typed buffer is converted to ASCII for the length of the buffer. For DPL services, refer to [Data Conversion For DPL Services](#).

6.1.2.3 Data Conversion for X_OCTET/CARRAY Typed Buffers

No data conversion is performed on these typed buffers. The remote host application or the ATMI application converts the data fields in the record, including all numeric and ASCII conversions.

These typed buffers are used when the data record cannot be described or converted using one of the other *strong* type buffers. *Strong* means Oracle Tuxedo Mainframe Adapter for SNA can understand all data fields and perform the required data conversion.

These typed buffers are also options when the remote service expects many styles of data aggregation (multiple record types), because the `OUTBUFTYPE` parameter is limited to one `type:subtype`.

6.1.2.4 Data Conversion for VIEW/VIEW32/X_C_TYPE/X_COMMON Typed Buffers

A subtype is required for these typed buffers. The subtype is the name of the view that describes the remote host record. The remote host record is converted to an ATMI

typed buffer. The compiled VIEW file is used to perform the data and ASCII conversion on the host record. By default, the host record is a COBOL data aggregate and the converted typed buffer is mapped by the ATMI application using a C structure. See the [Translation Rules for VIEW Data Types](#) section for more conversion options.

6.1.2.5 Data Conversion for FML/FML32 Typed Buffers

A subtype is required for these typed buffers. The subtype is the name of the view that describes the remote host record. The host record is converted to an FML buffer that is passed to the ATMI application.

By default, the host record is a COBOL record aggregate data type. The data is converted to a C structure, including ASCII conversion, using the compiled VIEW file. This data is then converted to an FML buffer using the field definitions associated with the VIEW.

6.1.2.6 Data Conversion for RECORD Typed Buffers

A subtype is required for this typed buffer. The subtype is the name of the RECORD that describes the remote host record. The remote host record is converted to an ATMI typed buffer. The compiled RECORD file is used to perform the data and ASCII conversion on the host record. By default, the host record is a COBOL data aggregate and the converted typed buffer is mapped by the ATMI application using a RECORD type.

6.1.3 Data Conversion For DPL Services

The Oracle Tuxedo Mainframe Adapter for SNA system supports remote CICS services as *Distributed Program Link (DPL)* programs. The DPL support is performed as if the ATMI service is a peer CICS/ESA service.

In a DPL program, the application is protected from all *Distributed Transaction Processing (DTP)*. The DPL application is a request/response service, with all data communication performed by the passing of a COMMAREA.

A basic DPL request API looks like this:

```
EXEC CICS LINK  
  
    PROGRAM ()  
    DATALENGTH ()  
    LENGTH ()  
    COMMAREA ()
```

In the preceding example, the requester sends the COMMAREA of DATALENGTH size and the receiving application receives the COMMAREA data contents in a buffer the size of LENGTH. The DATALENGTH size might be smaller than the LENGTH size, but the requester expects and receives the response in the original COMMAREA buffer the size of LENGTH.

The difference between a DPL program and an ATMI service is that a receiving ATMI service can resize a reply buffer, while the DPL program expects a reply buffer of a designated size. Also, an ATMI requester can receive a resized buffer in a buffer different from the original reply buffer.

The Oracle Tuxedo Mainframe Adapter for SNA software performs the manipulation described in the following subsections to smoothly adjust to the requirements of both types of applications.

- [DPL Requests Originating From an ATMI Application](#)
- [DPL Requests Originating From a CICS DPL](#)

6.1.3.1 DPL Requests Originating From an ATMI Application

The Oracle Tuxedo Mainframe Adapter for SNA software must determine what size `COMMAREA` the remote DPL service is expecting because no corresponding `LENGTH` parameter exists on an ATMI request.

To determine the `LENGTH` value for a DPL request, the software uses the larger potential size of the `INBUFTYPE` or the `OUTBUFTYPE` parameter definitions, as described in the following table.

The remote DPL application receives a buffer of `LENGTH` size and returns a buffer of `LENGTH` size.

Table 6-1 DPL Request LENGTH Calculation

INBUFTYPE or OUTBUFTYPE	LENGTH CALCULATION
STRING/X_OCTET/CARRAY	For these typed buffers, the <code>INBUFTYPE</code> parameter definition is used to determine the <code>LENGTH</code> and then add any null character.
VIEW/VIEW32/X_COMMON/X_C_TYPE	<code>LENGTH</code> is the maximum size of the <code>VIEW</code> file. This calculation takes in the potential size of both the C structure and the target record.
FML/FML32	The maximum size of the <code>VIEW</code> file. This calculation takes in the potential size of both the C structure, and the target record. The length of the FML buffer is not taken into account.
XML	When returning XML typed data, a DPL program may resize an XML typed reply buffer by appending null characters to the end of the XML string. To prevent errors in length of the reply buffer for XML typed data, any null characters appended to the XML string are removed before the data is returned.
RECORD	For this typed buffer, the <code>INBUFTYPE</code> parameter definition is used to determine the <code>LENGTH</code> .

If no `LENGTH` can be determined, then the largest value allowed by the CICS application is allocated.

6.1.3.2 DPL Requests Originating From a CICS DPL

The Oracle Tuxedo Mainframe Adapter for SNA software receives a `LENGTH` value and `COMMAREA` data of `DATALLENGTH` size from the requesting CICS DPL. The software allocates a buffer of `LENGTH` size and moves the `COMMAREA` data into this buffer before performing the conversions.

6.2 Translation Rules for VIEW Data Types

The following sections on default data translation rules provide suggestions to help you:

- [Develop VIEW definitions for input and output buffers and records](#)

- Understand how string data and numeric data are treated with the VIEW data type
- The following table lists VIEW data translation rules.

Table 6-2 VIEW Data Translation Rules

Field Type	Translation Rules
CARRAY	Passed without translation as sequences of bytes.
STRING and CHAR	Translated from ASCII to EBCDIC (If needed. Refer to "Translation Rules for Strings.")
SHORT	Translated to S9(4)COMP
LONG	Translated to S9(9)COMP
FLOAT	Translated to COMP-1
DOUBLE	Translated to COMP-2



Note:

The ATMI platform provides a field type named `dec_t` that supports decimal values within VIEWS. The Gateway translates these fields into machine independent representations of packed decimals. For example, `dec_t(m,n)` becomes `S9(2*m-(n+1))V9(n) COMP-3`. Therefore, a decimal field with a size of 8, 5 corresponds to `S9(10)V9(5) COMP-3`.

The following table lists the translation rules between C and IBM Z data types.

Table 6-3 Data Translation Rules between C and IBM Z Data Types

Remote Data Type	Description	View Field Type/Length
PIC X(n)	Alpha-numeric Characters	string / n
PIC X	Single Alpha-numeric Character	char
PIC X(n)	Raw Bytes	carray / n
PIC X	Single Numeric Byte	carray / 1
PIC S9(4) COMP	32-bit Integer	short
PIC S9(9) COMP	64-bit Integer	long
COMP-1	Single-precision Floating Point	float
COMP-2	Double-precision Floating Point	double
PIC S9((m+(n+1))/2)V9(n) COMP-3	Packed Decimal	dec_t / m,n

- [String Considerations](#)
- [Converting Numeric Data](#)

6.2.1 String Considerations

When you create VIEW definitions for input and output records that are used by CICS/ESA applications, do not specify an extra position for the terminating null characters that are used in string fields. For example, when a CICS/ESA application program expects 10 characters in an input record, specify 10 for that field, not 10 plus 1.

 **Note:**

Although Oracle Tuxedo Mainframe Adapter for SNA software does not require strings to be null-terminated, it respects null termination. When the Gateway software detects a null (zero) character within a string, it does not process any subsequent characters. To pass full 8-bit data that contains embedded null values, use a CARRAY type field or buffer.

The character set translations performed by Oracle Tuxedo Mainframe Adapter for SNA are fully localizable, in accordance with the X/Open XPG Portability Guides. ASCII and EBCDIC translations are loaded from message files. Refer to the [Translation Rules for Strings](#) section for more information.

The Oracle Tuxedo Mainframe Adapter for SNA software contains default behaviors that should meet the requirements of most English-language applications. However, you may find it necessary to customize the translation. Refer to the [Translation Rules for Strings](#) section for more information.

6.2.2 Converting Numeric Data

You can convert numeric data into different data types easily, if you specify enough range in the intermediate and destination types to handle the maximum value needed.

For example, you can convert a Field Manipulation Language (FML) field of double into a packed decimal field on the remote target system by specifying an appropriate `dec_t` type VIEW element.

You can also convert numeric values into strings. For example, while FML buffers do not directly support the `dec_t` type, you can place decimal values in string fields and map these to `dec_t` fields within VIEW definitions.

6.3 Translation Rules for RECORD Data Types

The data in RECORD buffer is always in original format - COBOL internal representation. For more information, see [Data Type and Conversion](#) in *Programming an Oracle Tuxedo ATMI Application Using FML*.

6.4 Translation Rules for Strings

When planning the interaction between the ATMI platform and host applications, consideration must be given to the programming languages in which the applications are written. A character string is represented differently in the COBOL language than

in the C language and associated ATMI platform VIEW buffer. The following listing demonstrates the three ways that the same two strings are coded (`string1` and `string2`).

Listing Three Representations of Strings

C Structure:

```
struct text
{
    char    rbufsize[5];
    char    testnum[2];
    char    sendnum;
    char    sysid[4];
    char    textfld[10];
    char    string1[10];
    char    string2[16];
};
```

VIEW text

#type	cname	fbyname	count	flag	size	null
char	rbufsize	-	5	-	-	-
char	testnum	-	2	-	-	-
char	sendnum	-	1	-	-	-
char	sysid	-	4	-	-	-
char	textfld	-	10	-	-	-
string	string1	-	1	-	10	-
string	string2	-	1	-	16	-

END

COBOL Record

```
01 TEXT.
   05 RBUFSIZEPIC X(5) .
   05 TEXTNUMPIC X(2) .
   05 SENDNUM PIC X.
   05 SYSID PIC X(4) .
   05 STRING1 PIC X(9) .
   05 STRING2PIC X(15) .
```

The listing shows that, in the C structure and VIEW buffer, the sizes of `string1` and `string2` are represented as 10 and 16 characters, respectively. However, in the COBOL record, the sizes are 9 and 15 characters, respectively. This incompatibility can cause code misalignment between C and COBOL programs if not anticipated in the source code.

To avoid such incompatibilities, the Gateway provides a software option to control the mapping of string data between C and COBOL applications. This option enables you to automatically compensate for the differences in null termination and padding characteristics of the two languages.

**Note:**

The option affects string fields in the ATMI platform VIEW buffers only. STRING buffers are not affected by this switch.

- [Setting the Option to Perform String Transformation](#)
- [Code Page Translation Tables](#)

6.4.1 Setting the Option to Perform String Transformation

To set the string transformation option, use the `CLOPT` parameter when you configure the Gateway server (GWSNAX) definition in the `UBBCONFIG` file. If you set the `-t` option of the `CLOPT` parameter to one of the values listed in the following table, the Gateway performs the corresponding string transformation. Use the following syntax format:

```
CLOPT="-- -t {number} [-c {value}]"
```

In this parameter, arguments and options are defined in the following way:

CLOPT

specifies the ATMI parameter which enables you to provide command-line options in a server definition.

`--` marks the end of system-recognized arguments and the start of arguments passed to a subroutine within the server. This option is required if you supply application-specific arguments, such as the `-t` option, to the server.

`-t` is the Oracle Tuxedo Mainframe Adapter for SNA option to establish C-to-COBOL string transformation.

`{number}` indicates the type of string transformation the Gateway performs as shown in the following table.

**Note:**

If you do not set the `-t` option of the `CLOPT` parameter in your server definition, by default the Gateway performs no string transformation.

Table 6-4 C to COBOL String Transformation

CLOPT -t Parameter Value	ATMI Application Language	Host Application Language
Not Set	No string transformation established	
1	C	COBOL
2	COBOL	C
3	C	C
4	COBOL	COBOL

These options function in the following ways:**Option value 1:**

For outbound messages to the host, C string fields are converted to COBOL string fields. All available characters, up to the defined length of the string and beginning with the null character, are converted to spaces and the length of the field is reduced by one.

For inbound messages from the host, COBOL string fields are converted to C string fields. When the -c option is not specified, the trailing blanks 0x40 are converted to null characters (zero value), and the field length is increased by one; when the -c option is specified, the trailing characters encoded in the -c value are converted to null characters (zero value), and the field length is increased by one.

Option value 2:

For outbound messages to the host, COBOL string fields are converted to C string fields, trailing blanks are converted to null characters (zero value) and the length of the field is increased by one.

For inbound messages from the host, C string fields are converted to COBOL string fields. When the -c option is not specified, all available characters, up to the defined length of the string and beginning with the null character, are converted to spaces (0x40) and the length of the field is reduced by one; when the -c option is specified, all available characters, up to the defined length of the string and beginning with the null character, are converted to the value specified by the -c option and the length of the field is reduced by one.

Option values 3 and 4:

No string transformations are made between programs written in compatible languages.

Example:

The following example of a server definition uses the switch to establish string transformations between an ATMI application written in C and a host application written in COBOL.

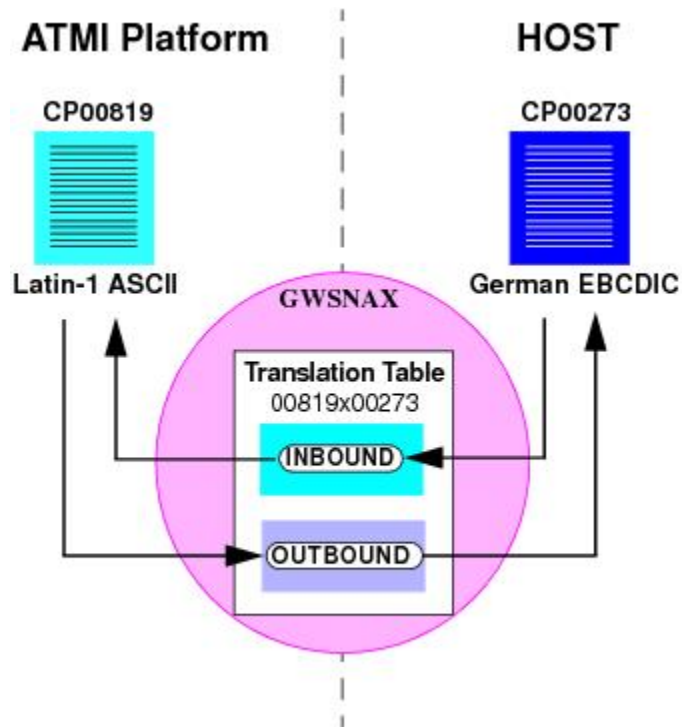
```
*SERVERS  
GWSNAX SRVGRP=GROUP1 SRVID=5 CLOPT="-A -- -t 1"
```

6.4.2 Code Page Translation Tables

The Oracle Tuxedo Mainframe Adapter for SNA software includes translation tables that enable conversions between ASCII character sets and EBCDIC character sets. The code pages provide 12 standardized tables to facilitate operations between ATMI applications using the Latin-1 ASCII code set (CP-00819) and host applications using a national language code set.

Each translation table consists of two mapping tables, one for outbound conversions (ATMI platform-to-host) and one for inbound conversions (host to ATMI platform). You do not have to specify the direction of a translation. You only need to determine the national language in which the host application is written. The following figure illustrates code page translation.

Figure 6-1 Oracle Tuxedo Mainframe Adapter for SNA Code Page Translation



The figure demonstrates how an ATMI application using the Latin-1 ASCII code page CP-00819 character set operates with a host application using German EBCDIC code page CP-00273. The Oracle Tuxedo Mainframe Adapter for SNA translation table 00819x00273 provides both the inbound and outbound conversions.

- [Specifying a Translation Table](#)
- [How the Translation Tables Work](#)

6.4.2.1 Specifying a Translation Table

To designate the translation table for your applications, make an entry in the ATMI platform `DMCONFIG` file definition for each remote domain. Use the `CODEPAGE` parameter with the following format:

For example:

```
*DM_REMOTE_DOMAINS
ORACLE TYPE=SNAX CODEPAGE="cpname"
```

In this parameter, `cpname` identifies the translation table for the remote domain, from the following table. It must be enclosed by double quotes.

The following table lists the translation tables provided with Oracle Tuxedo Mainframe Adapter for SNA software.

Table 6-5 Oracle Tuxedo Mainframe Adapter for SNA Code Page Translation Tables

Country	File Name	ASCII Code Set	EBCDIC Code Set
N/A	none	No translation	No translation
ATMI platform default ¹	tuxedo	ATMI ASCII	ATMI EBCDIC
United States	00819x00037	CP-00819 ²	CP-00037
Great Britain	00819x00285	CP-00819	CP-00285
France	00819x00297	CP-00819	CP-00297
Portugal	00819x00860	CP-00819	CP-00860
Spain	00819x00284	CP-00819	CP-00284
Belgium	00819x00500	CP-00819	CP-00500
Germany	00819x00273	CP-00819	CP-00273
Finland	00819x00278	CP-00819	CP-00278
Sweden	00819x00278	CP-00819	CP-00278
Italy	00819x00280	CP-00819	CP-00280
Latin-1	00819x01047	CP-00819	CP-01047
Latin-2	00912x00870	CP-00912	CP-00870

6.4.2.2 How the Translation Tables Work

The Oracle Tuxedo Mainframe Adapter for SNA translation tables are based on IBM-defined code sets. At start up, the Gateway loads a translation table for each remote domain.

You can modify any of the tables to suit your application translation needs, except the default ATMI tables, which are hard-coded. Refer to Appendix C, [Code Page Translation Tables](#) for detailed table contents. You must restart the Gateway to change any translation table definitions.



Note:

Replicas of the default ATMI translation tables are included with your product software. These tables are provided for you to modify, if desired. They are not the actual default tables. You cannot modify the default ATMI tables.

The Oracle Tuxedo Mainframe Adapter for SNA translation tables are located in the following sub-directory:

```
$TUXDIR/udataobj/codepage
```

If no `CODEPAGE` specification is made for a remote domain, the Oracle Tuxedo Mainframe Adapter for SNA software uses the ATMI default translation tables. If the software cannot find the translation table file, it generates a message `2241:ERROR Unable to access codepage table` with a reason code and the Gateway fails to start. Refer to this message in Appendix B, [Error Messages](#) for explanations of the reason codes.

The following listing depicts entries defining one local domain (CIXA) and two remote domains (CISA and IMSA). In all cases, the assumption is made that the local domain uses

ASCII code page CP-00819. In the example, the two remote domains use the German and French EBCDIC code pages CP-00273 and CP-00297, respectively.

Listing Code Page Definition Example

```
# DMCONFIG
*DM_LOCAL_DOMAINS
CIXA TYPE=SNAX
*DM_REMOTE_DOMAINS
CISA TYPE=SNAX CODEPAGE="00819X00273"
IMSA TYPE=SNAX CODEPAGE="00819X00297"
```

6.5 Multi-byte Character Set (MBCS) Support

Besides single-byte character set (SBCS) character set translation, TMA SNA supports multi-byte character set (MBCS) translation to meet Asian customer's requirement. This translation does not depend on translation tables under directory `$TUXDIR/udataobj/codepage`, instead, it is based on ICU libraries, which contains more than 200 different SBCS and MBCS character sets.

To use this translation, `CODEPAGE` in `DMCONFIG DM_REMOTE_DOMAINS` should be specified as following format:

```
CODEPAGE="ASCII_CHAR_SET:EBCDIC_CHAR_SET"
```

`ASCII_CHAR_SET` is for local domain and `EBCDIC_CHAR_SET` is for remote domain. They are separated by a colon.

Both `ASCII_CHAR_SET` and `EBCDIC_CHAR_SET` must be available character sets known by ICU. A utility `uconv` can be used to retrieve relevant information of these character sets.

For example, local domain uses ASCII Chinese code set `gb2312-1980`, and remote domain uses EBCDIC Chinese code set `ibm-1388`:

```
# DMCONFIG
*DM_REMOTE_DOMAINS
CICA TYPE=SNAX CODEPAGE="gb2312-1980:ibm-1388"
```

6.6 ART for CICS Integration Consideration

Tuxedo ATMI applications are mostly in C while ART for CICS applications are in COBOL; though both C and COBOL use the same VIEW32 definitions, there are two major differences.

- In C, the size of long type is 8 bytes; in COBOL, the size of long type is usually 4 bytes.
- C has an alignment requirement while COBOL does not.

Consequently, when it comes to ART for CICS integration, viewc32 needs two special compile flags. For example, on Linux,

```
CFLAGS="-m32 -fpack-struct=1" viewc32 myview.v
```

"-m32" is specified to use 32-bit long for COBOL "S9(9) COMP", and "-fpack-struct=1" is specified for non-alignment for COBOL. Ensure that only GWSNAX can use the VIEW32 binary (other C applications cannot use it).

In terms of ART for CICS integration, GWSNAX -t parameter can be omitted or set as "-t 4", for the null terminator is not required.

7

APPC/IMS Programming Considerations

This section is intended for application programmers who implement and integrate ATMI platform and host enterprise applications using Application Program-to-Program Communication/Information Management System (APPC/IMS) programs. The application programmer in the *IMS* environment can use implicit or explicit IMS programming techniques.



Note:

All references to ATMI files, functions, and documentation apply to Tuxedo files, functions, and documentation.

This chapter contains the following topics:

- [APPC/IMS Overview](#)
- [APPC/IMS Programming](#)

7.1 APPC/IMS Overview

APPC/IMS allows application programs using APPC protocols to enter IMS transactions from LU 6.2 devices supporting APPC. APPC/IMS also provides an environment that enables remote LU 6.2 devices to enter IMS local and remote transactions. In this environment, IMS application programs can insert transaction output to LU 6.2 devices without requiring coding changes to existing application programs and new application programs can make full use of existing LU 6.2 facilities. Applications enter transactions using an implicit or explicit Application Programming Interface (API).

- [Implicit API](#)
- [Explicit API](#)

7.1.1 Implicit API

The implicit API can be a useful simplification for many applications. While it does not provide all the existing LU6.2 capabilities, this API provides additional functions, such as message queuing and automatic asynchronous message delivery.

Using the IMS application programming base with the implicit API, you can write transactional applications that do not have Common Programming Interface for Communications (CPI-C) calls. IMS generates all the CPI-C calls for you. The application interaction is strictly with the IMS message queue.

The implicit API accesses an APPC conversation indirectly. It uses the standard DL/I calls (GU, ISRT, PURG) to send and receive data. The implicit API allows non-LU 6.2 specific applications to use LU 6.2 transactional protocols, using new and changed DL/I calls (CHNG, INQY, SETO).

The implicit API creates asynchronous LU 6.2 output by using alternate PCBs referencing LU 6.2 destinations. The DL/I CHNG call can supply parameters to specify an LU 6.2 destination. Default values substitute for omitted parameters.

An application program can use the implicit API to retrieve the current conversation attributes, such as the conversation type (basic or mapped), the sync_level, and whether it is asynchronous or synchronous.

7.1.2 Explicit API

An IMS application program can use the explicit API to issue the CPI-C calls directly. The explicit API is useful with remote LU 6.2 systems that have incomplete LU 6.2 implementations, or that are incompatible with the IMS implicit API support.

The explicit API can be used by any IMS application program to access an APPC conversation directly. IMS resources are available to the CPI-C driven application program only if the application issues the APSB (Allocate_PSB) call. The CPI-C driven application program must use the CPI-RR `SRRCMIT` and `SRRBACK` verbs to initiate an IMS sync point or backout.

7.2 APPC/IMS Programming

The Oracle Tuxedo Mainframe Adapter for SNA system supports non-transactional and transactional IMS servers using either the implicit APPC support for IMS or the explicit APPC interface using APPC/MVS calls from a user application. Any IMS program that gets messages from, and puts messages into, the IMS message queue can be used without change as either a client or server.

To use the implicit APPC capabilities of IMS, you must modify the `APPCMxx` file in the `SYS1.PARMLIB` library provided with your Oracle Tuxedo Mainframe Adapter for SNA software. The configuration parameters in this file associate the LU with the IMS scheduler. You must identify the LU representing the application name used by Oracle Tuxedo Mainframe Adapter for SNA to access the IMS region and the IMS system ID which provides scheduling for inbound requests. Be sure to discuss with mainframe support personnel the changes you make to the `APPCMxx` file.

- [Non-Transactional Application Programming](#)
- [Transactional Application Programming](#)
- [Sample Transaction Programs](#)

7.2.1 Non-Transactional Application Programming

The following listing is an example of a non-transactional program. In this example, the VTAM application major node is designated to be `MVSLUO1` and the scheduling facility is designated to be the IMS control region `IVP4`.

Listing APPCM File in SYS1.PARMLIB Library (Example Only)

```
SYS1.PARMLIB (APPCMxx)

LUADD ACBNAME (MVSLUO1) BASE TPDATA (SYS1.APPCTP),
SCHED (IVP4),

SIDEINFO DATASET (SYS1.APPCSI)
```

```

SYS1.VTAMLST(MVSLU01)

MVSLU01 APPL ACBNAME=MVSLU01,          ACBNAME FOR APPC          C
          APPC=YES,                    C
          AUTOSES=0,                   C
          DDRAINL=NALLOW,             C
          DLOGMOD=APPCHOST,           C
          DMINWNL=3,                  C
          DMINWNR=3,                  C
          DRESPL=NALLOW,             C
          DSESLIM=6,                  C
          LMDENT=19,                  C
          MODETAB=APPCTAB,            C
          PARSESS=YES,                C
          SECACPT=CONV,               C
          SRBEXIT=YES,                C
          VPACING=1

```

The job that starts the IMS subsystem should have the APPC parameter set to Y. The example in the following listing illustrates such a job, but is not intended to be used under actual conditions. Use your own custom job for starting IMS.

Listing IMS Subsystem Start Job (Example Only)

```

PROC RGN=2000K,SOUT=A,DPTY='(14,15)',
      SYS=,SYS1=,SYS2=,
      RGSUF=IV1,PARM1=APPC=Y,PARM2=,APPLID1=IMS61CR1,AOIS=R IEFPROC EXEC
PGM=DFSMVRC0,DPTY=&DPTY,
      REGION=&RGN,
      PARM='CTL,&RGSUF,&PARM1,&PARM2,&APPLID1,&AOIS'

*
*
* THE MEANING AND MAXIMUM SIZE OF EACH PARAMETER
* IS AS FOLLOWS:
*
***** CONTROL REGION SPECIFICATIONS *****
*****
*   RGSUF   XXX   EXEC   PARM   DEFAULT   BLOCK   SUFFIX   FOR
*           MEMBER   DFSPBXXX.
*****
*
* PARM1 , PARM2 PARAMETERS BOTH ARE USED TO SPECIFY
* CHARACTER STRINGS THAT CONTAIN IMS KEYWORD
*
* PARAMETERS. I.E. PARM1='AUTO=Y,PST=222,RES=Y'
*
*
* APPC     X           Y = ACTIVATE APPC/IMS
*           N           N = DO NOT ACTIVATE APPC/IMS

```

7.2.2 Transactional Application Programming

The following listing is an example of a transactional VTAM program. The inclusion of the LU definition `SYNCLVL=SYNCPT` (shown in bold) makes the program transactional.



Note:

You must include the `ATNLOSS=ALL` parameter value whenever you use the `SYNCLVL=SYNCPT` definition.

Listing Sample VTAM LU Definition

```
MVSLU01 APPL          ACBNAME=MVSLU01,      ACBNAME FOR APPC      C
                      APPC=YES,              C
                      AUTOSES=0,            C
                      DDRAINL=NALOW,        C
                      DLMOD=APPCHOST,       C
                      DMWNL=5,              C
                      DMINWNR=5,            C
                      DRESPL=NALLOW,       C
                      DSESLIM=10,          C
                      LMDENT=19,           C
                      MODETAB=APPCTAB,     C
                      PARSESS=YES,         C
                      SECACPT=CONV,        C
                      SRBEXIT=YES,         C
                      SYNCLVL=SYNCPT,      C
                      ATNLOSS=ALL,         C
                      VPACING=1
```

7.2.3 Sample Transaction Programs

The following Oracle Tuxedo Mainframe Adapter for SNA transactional test programs are installed in the ATMI platform installation in the sub-directory `TMA/sna/simpapp`:

- `simpims.c` is a simple ATMI client used to invoke both the sample IMS server programs. It takes a data string and service name as inputs. It invokes the service and passes the input data string.
- `IMPIMSSV.cbl` is a simple IMS echo server. It reads data from the IMS message queue and writes the same data in response. It is intended to be used as an implicit SNA example.
- `EXPIMSSV.c` is an IMS server transaction using explicit CPI-C calls. It is written for sync level 2 use. The program uses an IBM sample database, `IVPDB2`. The program displays, adds, and deletes records from the database, based on an input string. Sample input strings are documented in the source.
- `BEAWTOR.asm` is an assembler subroutine used by `EXPIMSSV.c` to write messages to the MVS console log.

8

Channel/Container Support in DPL

This chapter contains the following topics:

- [Channel / Container Overview](#)
- [ATMI Server Programming](#)
- [Configurations](#)

8.1 Channel / Container Overview

The channel and container model introduces a new approach that provides an easy and flexible mechanism for exchange of large volumes of structured parameter data between CICS programs.

Containers are named blocks of data designed for passing information between programs. Programs can pass any number of containers between each other. Containers grouped together in sets are called channels. A channel is analogous to a parameter list.

ART CICS uses `FML32` buffer type to carry channel data, and `CARRAY` to carry container data. Channel and container decode, encode, and data alignment are added to `TMA GWSNAX`. `GWSNAX` can be integrated with ART CICS to communicate between ART CICS and mainframe CICS, as well as between local Tuxedo and CICS programs.

8.2 ATMI Server Programming

ATMI server can only use FML32 APIs to manipulate channel / container according to following approximate mapping:

Table 8-1 CICS Container and FML32 APIs Mapping

CICS Container Commands	CICS Container Commands
PUT	Fadd32 () or Fchg32 ()
GET	Fget32 () or Ffind32 ()
DELETE	Fdel32 ()
STARTBROWSE GETNEXT ENDBROWSE	A loop based on Foccur32 ()

Meanwhile, five FML32 field macros should be included in ATMI server code; all of them belong to a given FML32 definition:

```
#define CX_PROGRAM ((FLDID32)231327331)
#define CX_TRANSID ((FLDID32)231327333)
#define CX_CHANNEL_NAME ((FLDID32)231329593)
#define CX_CONTAINER_NAME ((FLDID32)231329594)
#define CX_CONTAINER_DATA ((FLDID32)231329595)
```


These fields are used to map relevant CICS command options:

Table 8-2 CICS Command Options and FML32 Fields Mapping

CICS Command Options	Mapping FML32 Fields
PROGRAM of LINK	CX_PROGRAM
TRANSID of LINK	CX_TRANSID
CHANNEL of LINK or container commands	CX_CHANNEL_NAME
CONTAINER of container commands	CX_CONTAINER_NAME
FROM of PUT container INTO or SET of GET container	CX_CONTAINER_DATA
LENGTH of PUT or GET container	Length of CX_CONTAINER_DATA

8.3 Configurations

- [VIEW32 Definitions](#)
- [DMCONFIG](#)
- [Environment Variable](#)

8.3.1 VIEW32 Definitions

TMA SNA uses View32 definitions generated from copybook of containers for data conversion. Usually, every kind of container must correspond to a View32 definition of the same name but differs in the following aspects.

- TMA SNA container name is usually in uppercase while its corresponding View32 name is always in lowercase.
- Dash in TMA SNA container name needs to be converted to underscore.

For example, TMA SNA container name "CONT-1" corresponds to View32 name "cont_1".

Note:

In mainframe CICS, the action scope of container name is the current task; however, since it is represented by a VIEW32 in Oracle Tuxedo, the action scope expands to VIEW32. Consequently, Oracle Tuxedo cannot support different containers with the same name.

8.3.2 DMCNFIG

Both `INBUFTYPE` and `OUTBUFTYPE` must be set as `FML32` for services which use channel / container data type, but no `VIEW32` subtype is needed to specify because it will be automatically set as the container name.

8.3.3 Environment Variable

CHANNEL_COMPRESS_LEVEL

The environment variable is used to improve network transmission performance between GWSNAX and CRM when channel/container contains large amounts of data.

When it is set, GWSNAX or CRM compresses channel / container data before sending it to network. The value is an integer in the range of 0 to 9. Lower compression levels result in faster execution, but less compression. Higher levels result in greater compression, but slower execution.

For example: CHANNEL_COMPRESS_LEVEL=6

CHANNEL_WITHOUT_ART_FLDS

Besides the five FML32 fields mentioned in the above [ATMI Server Programming](#) section, GWSNAX provides the following three fields only for ART CICS integration.

```
#define CXMW_MESSAGE ((FLDID32)231327692) #define CX_USERID  
((FLDID32)231327345) #define CX_USERNAME ((FLDID32)231327376)
```

When this environment variable is set, the above three fields, which are useless in other scenarios, will not be added in FML32 buffer sent to queue.

For example: CHANNEL_WITHOUT_ART_FLDS=y

See Also:

- [ART for CICS Integration Consideration](#) in *Oracle Tuxedo Mainframe Adapter for SNA User's Guide*
- [GWSNAX](#) in *Oracle Tuxedo Mainframe Adapter for SNA User's Guide*

9

Inbound Transaction Routing

This chapter contains the following topics:

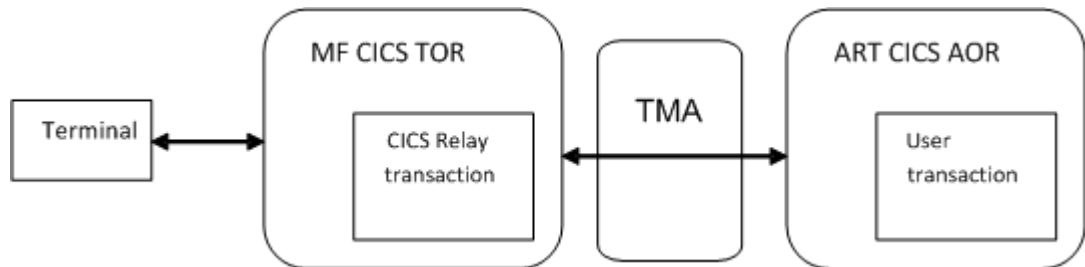
- [Overview](#)
- [Configuration](#)
- [Example\(s\)](#)

9.1 Overview

CICS transaction routing allows terminals connected to one CICS system to run with transactions in another connected CICS system. This means that you can distribute terminals and transactions around your CICS systems and still have the ability to run any transaction with any terminal.

This feature aims that TMA SNA supports inbound terminal initiated transaction routing in MF CICS/ART CICS integration, in such a scenario, MF CICS can only be TOR and ART CICS can only be AOR.

Figure 9-1 Inbound Transaction Routing Support Configuration



9.2 Configuration

Transactions can be triggered from a TOR terminal and run on AOR; the execution process is exactly as same as it running local.

- [MF CICS Configuration](#)
- [ARTCICS and TMA SNA Configuration](#)

9.2.1 MF CICS Configuration

1. Define terminals, alternatively, auto installed terminal can be used as well.
2. Do not specify program parameter for relevant transactions, instead, specify remote system for them.

3. Define connection session for TMA SNA as usual.

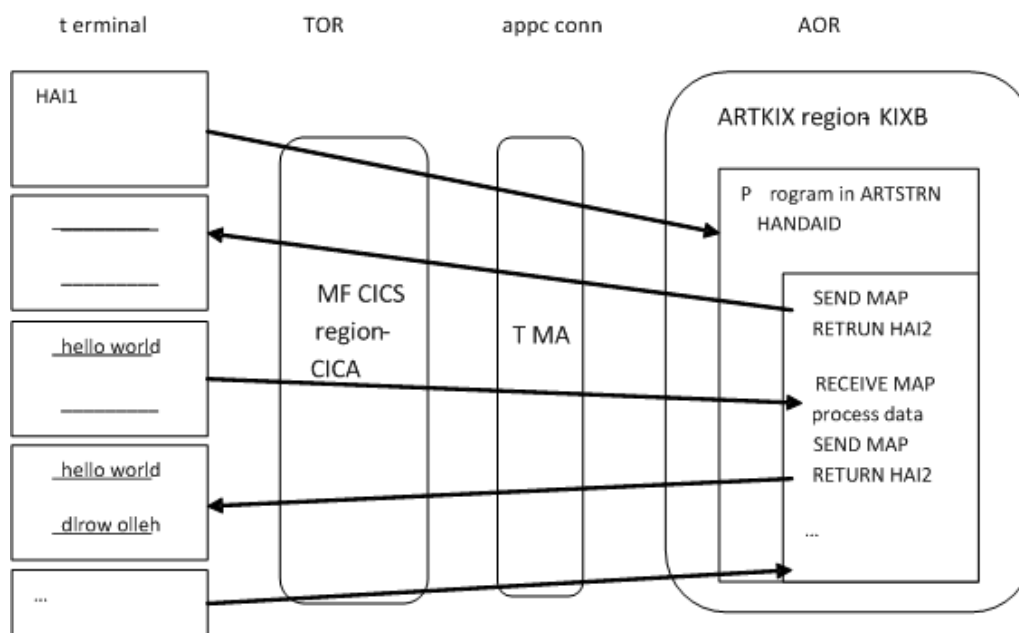
9.2.2 ARTCICS and TMA SNA Configuration

1. Configure ARTCICS servers and GWSNAX server in the UBBCONFIG file.
2. Create a DMCONFIG for TMA SNA, and define ARTSTRN exported services in the DM_EXPORT section, which corresponds to the TOR-invoked transactions. "CONV=Y" is mandatory here because ARTSTRN is a conversational server.
3. Define transactions/programs/mapsets in ARTCICS as usual.

9.3 Example(s)

The following figure shows a typical example.

Figure 9-2 Example 1



There is a TOR with terminals on mainframe, and an ART AOR on open system, they communicate each other via APPC connection provided by TMA SNA.

A pseudo conversation transaction HAI1 starts from a TOR terminal. This transaction is defined as a remote transaction on AOR, so TOR forwards the request to AOR. This causes the transaction to execute; the terminal and transaction exchange screen data via the transaction routing mechanism.

- [Configuration](#)

9.3.1 Configuration

1. Transactions on mainframe TOR

```
DEFINE TRANS (HAI1) GROUP (SIMPAPP) REMOTES (CR01)  
DEFINE TRANS (HAI2) GROUP (SIMPAPP) REMOTES (CR01)
```

 **Note:**

CR01 is a CICA connection, which points to TMA SNA LU

2. Services in TMA DMCONFIG

```
*DM_EXPORT  
"HAI1" CONV=Y  
"HAI2" CONV=Y
```

3. Transactions on ART AOR

```
HAI1;SIMPAPP;pg for simpapp;HANDAID  
HAI2;SIMPAPP;pg for simpapp;HANDAID
```

4. Program on ART AOR

```
HANDABD;SIMPAPP; pg for simpapp; COBOL;;ENABLED
```

 **Note:**

The relevant mapset definition is not mandatory in ART CI

CS.

10

TMA/SNA High Availability

TMA SNA supports full high availability connection between Oracle Tuxedo and CICS, with no single point failure. Both bidirectional load balancing and bidirectional failover are supported.

This chapter contains the following topics:

- [Outbound Call \(from Oracle Tuxedo to CICS\)](#)
- [Inbound Call \(from CICS to Oracle Tuxedo\)](#)

10.1 Outbound Call (from Oracle Tuxedo to CICS)

TMA SNA supports two types of outbound call infrastructures:

- [Single Connection and Multiple Links](#)
- [Multiple Connections and Multiple Links](#)

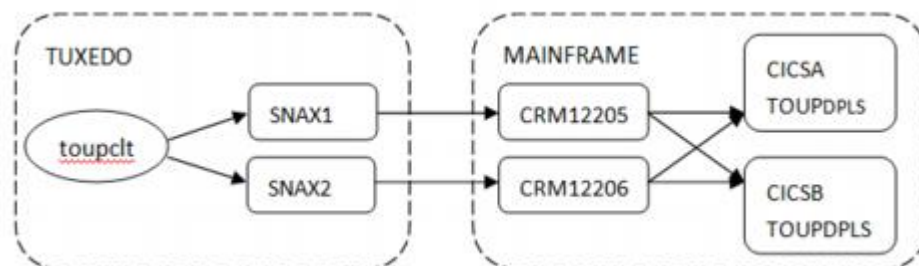
The term *connection* is used for the TCP connection between GWSNAX and CRM, and the term *link* is used for the SNA connection between CRM and CICS.

- [Single Connection and Multiple Links](#)
- [Multiple Connections and Multiple Links](#)

10.1.1 Single Connection and Multiple Links

As shown in the following figure, one Tuxedo client calls a remote service, which maps to a CICS DPL program. The DPL program runs in two CICS regions – CICS A and CICS B. Meanwhile, two GWSNAX/CRM pairs are configured to avoid any single point of failure. Each CRM connects to both CICS regions.

Figure 10-1 Outbound Call (Single Connection and Multiple Links)



In this scenario, the load balancing / failover can be accomplished via following four paths:

```
toupclt -> SNAX1 -> CRM12205 -> CICSA -> TOUPDPLS
toupclt -> SNAX1 -> CRM12205 -> CICSB -> TOUPDPLS
```

```
toupclt -> SNAX2 -> CRM12206 -> CICSA -> TOUPDPLS
toupclt -> SNAX2 -> CRM12206 -> CICSB -> TOUPDPLS
```

The following table lists all the configurations required for making outbound calls of single connection and multiple links.

Table 10-1 Configurations for Single Connection and Multiple Links (Outbound)

Machine	Component	Resource	Resource
TUXEDO	UBBCONFIG	*GROUPS	2 groups for 2 GWSNAX servers
		*SERVERS	2 GWADM/GWSNAX pairs
	DMCONFIG	*DM_LOCAL	2 local domains: SNAX1, SNAX2
		*DM_REMOTE	4 remote domains: CICSA1, CICSB1, CICSA2, CICSB2
		*DM_SNACRM	2 CRMs: CRM1, CRM2
		*DM_SNALINKS	4 LINKs: CICSA1, CICSB2, CICSA2, CICSB2
		*DM_IMPORT	4 remote homonymous services as 2X2 LACCESSPOINT / RACCESSPOINT combinations: DPLS
MAINFRAME	VTAM list	Major node	2 APPC LUs (for two CRMs)
	CICSA	Resource	2 CONNECTION/SESSION pairs for two LUs
			1 PROGRAM for DPLS
	CICSB	Resource	2 CONNECTION/SESSION pairs for two LUs
1 PROGRAM for DPLS			

- [Configuring UBBCONFIG](#)
- [Configuring DMCNFIG](#)
- [Configuring VTAM List](#)
- [Configuring Connections](#)
- [Programming for DPLS](#)

10.1.1.1 Configuring UBBCONFIG

In the Tuxedo UBBCONFIG file, configure the following:

1. Add two GROUP entries:
Listing Add Groups

```
*GROUPS
SNAX1
    LMID=SITE1 GRPNO=10
SNAX2
    LMID=SITE1 GRPNO=9
```

2. Add two GWADM and GWSNAX servers:

Listing Add Servers

```
*SERVERS
GWADM
    SRVGRP=SNAX1 SRVID=9
GWADM
    SRVGRP=SNAX2 SRVID=10

GWSNAX

    SRVGRP=SNAX1 SRVID=12
    RQADDR="SNAX1"
    REPLYQ=N RESTART=N GRACE=0
    CLOPT="-A -- -t 1"

GWSNAX

    SRVGRP=SNAX2 SRVID=13
    RQADDR="SNAX2"
    REPLYQ=N RESTART=N GRACE=0
    CLOPT="-A -- -t 1"
```

10.1.1.2 Configuring DMCONFIG

In the Tuxedo DMCONFIG file, configure the following:

1. Add two local domains:

Listing Add Local Domains

```
*DM_LOCAL
SNAX1
    GWGRP=SNAX1           # same as GWSNAX group in ubbconfig
    TYPE=SNAX
    DOMAINID="SNAX1"
    DMTLOGDEV="/home/wpwang/11TmaTest/sna_scm1_ha/SNAX1"
SNAX2
    GWGRP=SNAX2
    TYPE=SNAX
    DOMAINID="SNAX2"
    DMTLOGDEV="/home/wpwang/11TmaTest/sna_scm1_ha/SNAX2"
```

2. Add four remote domains:

Listing Add Remote Domains

```
*DM_REMOTE
CICSA1  TYPE=SNAX DOMAINID="CICSA1"
CICSA2  TYPE=SNAX DOMAINID="CICSA2"
CICSB1  TYPE=SNAX DOMAINID="CICSB1"
CICSB2  TYPE=SNAX DOMAINID="CICSB2"
```

3. Add two CRMs:

Listing Add CRM

```
*DM_SNACRM
CRM1
  SNACRMADDR="//wasa:12205"
  NWDEVICE="/dev/tcp"
  LDOM="SNAX1"
```

CRM2

```
  SNACRMADDR="//wasa:12206"
  NWDEVICE="/dev/tcp"
  LDOM="SNAX2"
```

4. Add two SNA stacks:

Listing Add SNA Stacks

```
*DM_SNASTACKS
CRMST1
  SNACRM="CRM1"
  STACKTYPE="VT210"
  LOCALLU="CRMLU06"      #NETNAME in VTAM CONNECTION definition
  LTPNAME="*"
  STACKPARMS="wasa"
CRMST2
  SNACRM="CRM2"
  STACKTYPE="VT210"
  LOCALLU="CRMLU07"      #NETNAME in VTAM CONNECTION definition
  LTPNAME="*"
  STACKPARMS="wasa"
```

5. Add four SNA links:

Listing Add SNA Links

```
*DM_SNALINKS
DEFAULT:
  MODENAME="SMSNA100"
  SECURITY="LOCAL"
  STARTTYPE="COLD"
  MAXSESS=8
  MINWIN=4
  MAXSYNCLVL=2
CICSA1 STACKREF="CRMST1" LSYSID="CR06" RDOM="CICSA1" RSYSID="CICS"
  RLUNAME="CICSA"
CICSB1 STACKREF="CRMST1" LSYSID="CR06" RDOM="CICSB1" RSYSID="CICS"
  RLUNAME="CICSB"
CICSA2 STACKREF="CRMST2" LSYSID="CR07" RDOM="CICSA2" RSYSID="CICS"
  RLUNAME="CICSA"
CICSB2 STACKREF="CRMST2" LSYSID="CR07" RDOM="CICSB2" RSYSID="CICS"
  RLUNAME="CICSB"
```

6. Define four services:

Listing Define Services

```

*DM_IMPORT
DEFAULT:
    RACCESSPOINT=CICSA1
    LOAD=50
    FUNCTION=DPL
SIMPDPDPL
    LACCESSPOINT=SNAX1
    RACCESSPOINT=CICSA1
    RNAME="TOUPDPLS"
    INBUFTYPE="STRING"
    OUTBUFTYPE="STRING"
SIMPDPDPL
    LACCESSPOINT=SNAX1
    RACCESSPOINT=CICSA2
    RNAME="TOUPDPLS"
    INBUFTYPE="STRING"
    OUTBUFTYPE="STRING"
SIMPDPDPL
    LACCESSPOINT=SNAX2
    RACCESSPOINT=CICSB1
    RNAME="TOUPDPLS"
    INBUFTYPE="STRING"
    OUTBUFTYPE="STRING"
SIMPDPDPL
    LACCESSPOINT=SNAX2
    RACCESSPOINT=CICSB2
    RNAME="TOUPDPLS"
    INBUFTYPE="STRING"
    OUTBUFTYPE="STRING"

```

10.1.1.3 Configuring VTAM List

Define two APPC LUs, for instance, define the LU CRMLU06 and CRMLU07, as shown below:

Listing Define APPC LUs

```

CRMMN3      VBUILD TYPE=APPL
CRMLU06    APPL ACBNAME=CRMLU06,
              APPC=YES,
              SYNCLVL=SYNCPT,
              AUTH=(ACQ,PASS),
              PARSESS=YES
CRMLU07    APPL ACBNAME=CRMLU07,
              APPC=YES,
              SYNCLVL=SYNCPT,
              AUTH=(ACQ,PASS),
              PARSESS=YES

```

10.1.1.4 Configuring Connections

Define two CONNECTION/SESSION pairs for each LU. One CONNECTION/SESSION pair is used to connect to one CICS region. In the following listing, a CONNECTION/SESSION pair is defined:

Listing Define Connections

```
ADD GROUP (BEACON1) LIST (BEALST)

DEFINE CONNECTION (CR06) GROUP (BEACON1)
  DE (BEA TMA SNA EXAMPLE RDO CONNECTION)
  ACCESSMETHOD (VTAM)          PROTOCOL (APPC)
  NETNAME (CRMLU06)
  ATTACHSEC (LOCAL)           AUTOCONNECT (NO)

DEFINE SESSION (CR06S)          GROUP (BEACON1)
  CONNECTION (CR06)
  DE (SESSION EXAMPLE)
  PROTOCOL (APPC)              AUTOCONNECT (YES)
  MODENAME (SMSNA100)         MAXIMUM (10, 5)

DEFINE CONNECTION (CR07)          GROUP (BEACON1)
  DE (BEA TMA SNA EXAMPLE RDO CONNECTION)
  ACCESSMETHOD (VTAM)          PROTOCOL (APPC)
  NETNAME (CRMLU07)
  ATTACHSEC (LOCAL)           AUTOCONNECT (NO)

DEFINE SESSION (CR07S)          GROUP (BEACON1)
  CONNECTION (CR07)
  DE (SESSION EXAMPLE)
  PROTOCOL (APPC)              AUTOCONNECT (YES)
  MODENAME (SMSNA100)         MAXIMUM (10, 5)
```



Note:

Replace the value of GROUP, LIST, CONNECTION, NETNAME, SESSION, MODENAME, and MAXIMUM with your actual value.

Install resource definition in both CICS A and CICS B region, and enter commands to install resource definitions:

```
CEDA INSTALL GROUP (BEACON1)
```

10.1.1.5 Programming for DPLS

1. Compile the MIRRDPLC program using following JCL:

```
//COMPIL1 JOB CLASS=2,MSGCLASS=A,MSGLEVEL=(1,1),NOTIFY=&SYSUID
//PROCLIB JCLLIB ORDER=&SYSUID..SNASNT.JCL
```

```

/**
//  SET CBLDS=&SYSUID..SNASNT.COBOL
//  SET LDDS=&SYSUID..SNASNT.LOAD
/**
//STEP1 EXEC DFHYITVL,
//      PROG='TOUPDPLS',
//      SRCLIB=&CBLDS,
//      DSCTLIB=&CBLDS,
//      PROGLIB=&LDDS

```

2. Define DPL program in both CICS and CICSB regions:

```

DEFINE PROGRAM(TOUPDPLS)      GROUP(SNASNT)
      DESCRIPTION(BEA TMA SNA DPL CLIENT)
      LANGUAGE(COBOL)        DATALOCATION(ANY)

```

3. Install resource definition in both CICS and CICSB regions:
CEDA INSTALL GROUP(SNASNT)

10.1.2 Multiple Connections and Multiple Links

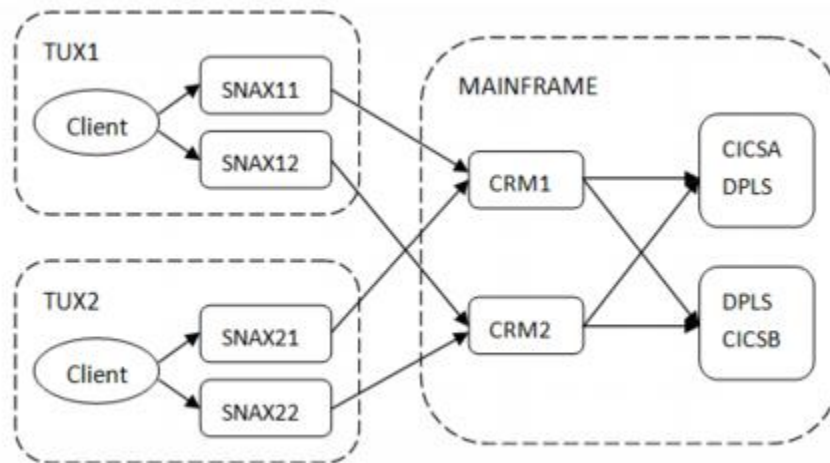
Unlike the single connection and multiple links, in this scenario, each CRM accepts two connections from GWSNAX in two separate Tuxedo domains. One domain serves as a master, and the other serves as a slave.

CRM supports both multiple connections and multiple links simultaneously.

 **Note:**

- Multiple connections is only supported by CRM 12cR1 or later.
- GWSNAXs connect to one CRM must be in different Tuxedo domains.

Figure 10-2 Multiple Connections and Multiple Links



The following table lists all the configurations required for making outbound calls of multiple connections and multiple links.

Table 10-2 Configurations for Multiple Connection and Multiple Links (Outbound)

Machine	File	Field	Content
TUX1	UBBCONFIG	*GROUPS	2 groups for 2 GWSNAX servers
		*SERVERS	2 GWADM/GWSNAX pairs
	DMCONFIG	*DM_LOCAL	2 local domains: SNAX11, SNAX12
		*DM_REMOTE	4 remote domains: CICSA1, CICSB1, CICSA2, CICSB2
		*DM_SNACRM	2 CRMs: CRM1, CRM2
	*DM_SNALINKS	4 LINKS: CICSA1, CICSB2, CICSA2, CICSB2	
TUX2	UBBCONFIG	*GROUPS	2 groups for 2 GWSNAX servers
		*SERVERS	2 GWADM/GWSNAX pairs
	DMCONFIG	*DM_LOCAL	2 local domains: SNAX21, SNAX22
		*DM_REMOTE	4 remote domains: CICSA1, CICSB1, CICSA2, CICSB2
		*DM_SNACRM	2 CRMs: CRM1, CRM2
	*DM_SNALINKS	4 LINKS: CICSA1, CICSB2, CICSA2, CICSB2	
MAINFRAME	VTAM list	Major node	2 APPC LUs (for two CRMs)
	CICSA	Resource	2 CONNECTION/SESSION pairs for two LUs
	CICSB	Resource	2 CONNECTION/SESSION pairs for two LUs

The configurations for multiple connections and multiple links are almost same as that of the single connection scenario and won't be explained again. You can refer to the preceding configuration steps in [Single Connection and Multiple Links](#). There are just some slight differences in the configuration of variables `IPCKEY`, `APPDIR`, `TUXCONFIG`, `TLOGDEVICE` in `UBBCONFIG`, and `DMTLOGDEV` in `DMCONFIG`.

10.2 Inbound Call (from CICS to Oracle Tuxedo)

TMA SNA supports two types of inbound call infrastructures:

- Single connection and multiple links
- Multiple connections and multiple links

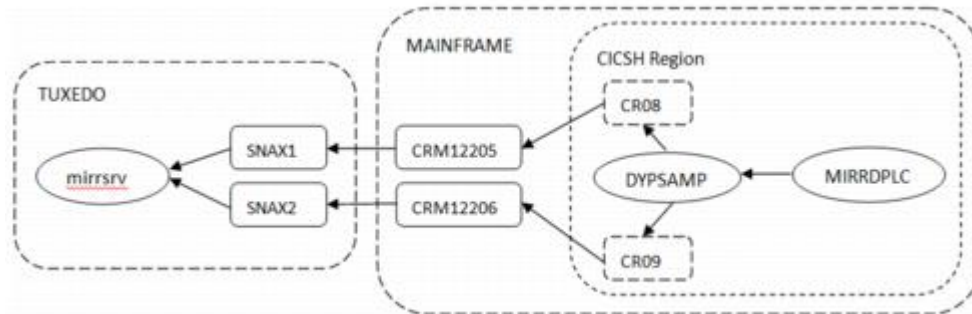
The term *connection* is used for the TCP connection between GWSNAX and CRM, and the term *link* is used for the SNA connection between CRM and CICS.

- [Single Connection and Multiple Links](#)
- [Multiple Connections and Multiple Links](#)

10.2.1 Single Connection and Multiple Links

As shown in the following figure, one CICS transaction links a remote program via DPL call, which maps to an Oracle Tuxedo service. The Tuxedo service is provided by one Tuxedo server and two GWSNAX/CRM pairs are configured to avoid any single point of failure.

Figure 10-3 Inbound Call (from CICS to Oracle Tuxedo)



The load balancing (round-robin) / failover can be accomplished via following two call paths:

```
MIRRDPLC->DYPSAMP -> CRM12205 -> SNAX1 -> mirrsrv
MIRRDPLC->DYPSAMP -> CRM12206 -> SNAX2 -> mirrsrv
```

Dynamic routing can be configured in CICS to dispatch the request to two CRMs with load balance and failover automatically, and the default dynamic routing program, `DFHDYP`, should be customized accordingly.

`DFHDYP` is a CICS-supplied, user-replaceable program for dynamic routing. For eligible DPL requests, it is invoked before a linked program is executed, to obtain the `SYSID` of the region where the link should be routed.

The following table lists all the configurations required for making inbound calls of single connection and multiple links.

Table 10-3 Configurations for Single Connection and Multiple Links (Inbound)

Machine	Component	Resource	Configuration
TUXEDO	UBBCONFIG	*GROUPS	2 groups for 2 GWSNAX servers
		*SERVERS	2 GWADM / GWSNAX pair
			1 application server (for inbound call)
	DMCONFIG	*DM_LOCAL	2 local domain: SNAX1, SNAX2
		*DM_REMOTE	2 remote domains: CICS1, CICS2
		*DM_SNACRM	2 CRM: CRM1, CRM2
		*DM_SNALINKS	2 LINK: CICS1, CICS2
	*DM_EXPORT	1 service in application server	

Table 10-3 (Cont.) Configurations for Single Connection and Multiple Links (Inbound)

Machine	Component	Resource	Configuration
MAINFRAME	VTAM list	Major node	2 APPC LUs (for two CRMs)
	CICSH	Resources	2 CONNECTION/SESSION pairs for two LUs. DFHDYP program: DYPSAMP TRANSACTION/PROGRAM for DPLC
		SIT	DFHDYP initialization parameter: DTRPGM=DYPSAMP

- [Configuring UBBCONFIG](#)
- [Configuring DMCONFIG](#)
- [Configuring VTAM List](#)
- [Configuring Connections](#)
- [Programming for DPLC](#)
- [Configuring CICS System Initialization Table](#)

10.2.1.1 Configuring UBBCONFIG

In the Tuxedo UBBCONFIG file, configure the following:

1. Add two GROUP entries:

Listing Add Groups

```
*GROUPS
SNAX1
    LMID=SITE1 GRPNO=4
SNAX2
    LMID=SITE1 GRPNO=5
```

2. Add two GWADM and GWSNAX servers:

Listing Add Servers

```
*SERVERS

GWADM

    SRVGRP=SNAX1 SRVID=11
GWADM
    SRVGRP=SNAX2 SRVID=12

GWSNAX
    SRVGRP=SNAX1 SRVID=14
    RQADDR="SNAX1"
    REPLYQ=N RESTART=N GRACE=0
    CLOPT="-A -- -t 1"

GWSNAX
    SRVGRP=SNAX2 SRVID=15
```

```
RQADDR="SNAX2"
REPLYQ=N RESTART=N GRACE=0
CLOPT="-A -- -t 1"
```

3. Add an application server:
Listing Add Server

```
*SERVERS
mirrorsrv
  SRVGRP=APPGRP SRVID=21 RQADDR="MIRROR" REPLYQ=Y CONV=N LMID=SITE1
```

10.2.1.2 Configuring DMCONFIG

In the Tuxedo DMCONFIG file, configure the following:

1. Add two local domains:
Listing Add Local Domains

```
*DM_LOCAL
SNAX1
  GWGRP=SNAX1           # same as GWSNAX group in ubbconfig
  TYPE=SNAX
  DOMAINID="SNAX1"
  DMTLOGDEV="/home/wpwang/11TmaTest/sna_scml_ha/SNAX1"
SNAX2
  GWGRP=SNAX2
  TYPE=SNAX
  DOMAINID="SNAX2"
  DMTLOGDEV="/home/wpwang/11TmaTest/sna_scml_ha/SNAX2"
```

2. Add two remote domains:
Listing Add Remote Domains

```
*DM_REMOTE
CICSA1 TYPE=SNAX DOMAINID="CICSA1"
CICSA2 TYPE=SNAX DOMAINID="CICSA2"
```

3. Add two CRMs:
Listing Add CRM

```
*DM_SNACRM
CRM1
  SNACRMADDR="//wasa:12205"
  NWDEVICE="/dev/tcp"
  LDOM="SNAX1"
CRM2
  SNACRMADDR="//wasa:12206"
  NWDEVICE="/dev/tcp"
  LDOM="SNAX2"
```

4. Add two SNA stacks:

Listing Add SNA Stacks

```
*DM_SNASTACKS
CRMST1
  SNACRM="CRM1"
  STACKTYPE="VT210"
  LOCALLU="CRMLU08" #NETNAME in VTAM CONNECTION definition
  LTPNAME="*"
  STACKPARMS="wasa"
CRMST2
  SNACRM="CRM2"
  STACKTYPE="VT210"
  LOCALLU="CRMLU09" #NETNAME in VTAM CONNECTION definition
  LTPNAME="*"
  STACKPARMS="wasa"
```

5. Add two SNA links:

Listing Add SNA Links

```
*DM_SNALINKS
DEFAULT:
  MODENAME="SMSNA100"
  SECURITY="LOCAL"
  STARTTYPE="COLD"
  MAXSESS=8
  MINWIN=4
  MAXSYNCLVL=2
CICSA1 STACKREF="CRMST1" LSYSID="CR08" RDOM="CICSA1" RSYSID="CICS"
  RLUNAME="CICSA"
CICSA2 STACKREF="CRMST2" LSYSID="CR09" RDOM="CICSA2" RSYSID="CICS"
  RLUNAME="CICSA"
```

6. Define one service:

Listing Define Service

```
*DM_EXPORT

DEFAULT:
  CONV=N

MIRROR
  RNAME="MIRRORSERV"
  INBUFTYPE="STRING"
  OUTBUFTYPE="STRING"
```

10.2.1.3 Configuring VTAM List

Define two APPC LUs, for instance, define the LU CRMLU08 and CRMLU09, as shown below:

Listing Define APPC LUs

```
CRMMN3          VBUILD TYPE=APPL
CRMLU08       APPL ACBNAME=CRMLU08,
```

```

                APPC=YES,
                SYNCLVL=SYNCPT,
                AUTH=(ACQ,PASS),
                PARSESS=YES
CRMLU09    APPL ACBNAME=CRMLU09,
                APPC=YES,
                SYNCLVL=SYNCPT,
                AUTH=(ACQ,PASS),
                PARSESS=YES

```

10.2.1.4 Configuring Connections

Define two CONNECTION/SESSION pairs for each LU. One CONNECTION/SESSION pair is used to connect to one CICS region. In the following listing, a CONNECTION/SESSION pair is defined:

Listing Define Connections

```

ADD GROUP (BEACON1)    LIST (BEALST)

DEFINE CONNECTION (CR08) GROUP (BEACON1)
    DE (BEA TMA SNA EXAMPLE RDO CONNECTION)
    ACCESSMETHOD (VTAM)    PROTOCOL (APPC)
    NETNAME (CRMLU08)
    ATTACHSEC (LOCAL)    AUTOCONNECT (NO)
DEFINE SESSION (CR08S) GROUP (BEACON1)
    CONNECTION (CR08)
    DE (SESSION EXAMPLE)
    PROTOCOL (APPC)    AUTOCONNECT (YES)
    MODENAME (SMSNA100)    MAXIMUM (10,5)
DEFINE CONNECTION (CR09) GROUP (BEACON1)
    DE (BEA TMA SNA EXAMPLE RDO CONNECTION)
    ACCESSMETHOD (VTAM)    PROTOCOL (APPC)
    NETNAME (CRMLU09)
    ATTACHSEC (LOCAL)    AUTOCONNECT (NO)
DEFINE SESSION (CR09S) GROUP (BEACON1)
    CONNECTION (CR09)
    DE (SESSION EXAMPLE)
    PROTOCOL (APPC)    AUTOCONNECT (YES)
    MODENAME (SMSNA100)    MAXIMUM (10,5)

```



Note:

Replace the value of GROUP, LIST, CONNECTION, NETNAME, SESSION, MODENAME, and MAXIMUM with your actual value.

Install resource definition in the CICS region, for example, CICS_H, and enter commands to install resource definitions:

```
CEDA INSTALL GROUP (BEACON1)
```

10.2.1.5 Programming for DPLC

1. Edit **DYPSAMP** to use specific connections defined before:

```
01 CONN-ST PIC S9(08) COMP.
01 RMT-SYS-1 PIC X(4) VALUE 'CR08'.
01 RMT-SYS-2 PIC X(4) VALUE 'CR09'.
```

```
LINKAGE SECTION.
```

2. Comment **SYSID** in **MIRRDPLC** to use dynamic routing:

```
EXEC CICS LINK
PROGRAM (REMOTE-SERVICE)
DATALENGTH (REQUEST-LEN)
LENGTH (LENGTH OF COMMAREA)
COMMAREA (COMMAREA)
SYNCONRETURN
*      SYSID (REMOTE-SYSID)
RESP (RESP-CODE)
END-EXEC
```

3. Compile the **MIRRDPLC** and **DYPSAMP** program using following JCL:

```
//COMPIL1 JOB CLASS=2,MSGCLASS=A,MSGLEVEL=(1,1),NOTIFY=&SYSUID
//PROCLIB JCLLIB ORDER=&SYSUID..SNASNT.JCL
//*
// SET CBLDS=&SYSUID..SNASNT.COBOL
// SET LDDS=&SYSUID..SNASNT.LOAD
//*
//STEP1 EXEC DFHYITVL,
//      PROG='MIRRDPLC',
//      SRCLIB=&CBLDS,
//      DSCTLIB=&CBLDS,
//      PROGLIB=&LDDS
//      STEP2 EXEC DFHYITVL,
//      PROG='DYPSAMP',
//      SRCLIB=&CBLDS,
//      DSCTLIB=&CBLDS,
//      PROGLIB=&LDDS
//*
```

4. Define programs in CICS:

```
DEFINE PROGRAM(DYPSAMP) GROUP(SNASNT)

DEFINE PROGRAM(MIRRDPLC)      GROUP(SNASNT)
DESCRIPTION(BEA TMA SNA DPL CLIENT)
LANGUAGE(COBOL)      DATALOCATION(ANY)
```

5. Install resource definition in the CICS region:

```
CEDA  INSTALL GROUP (SNASNT)
```

10.2.1.6 Configuring CICS System Initialization Table

Add the following definition in the CICS system initialization table, `CICSTS42.CICS.CICSH.SYSIN(DFHYSIPT)`:

```
*PLTPI=00                                DFHPLT Table Assembly (DL2)
DTRPGM=DYPSAMP
RLS=YES                                  record level sharing
```

10.2.2 Multiple Connections and Multiple Links

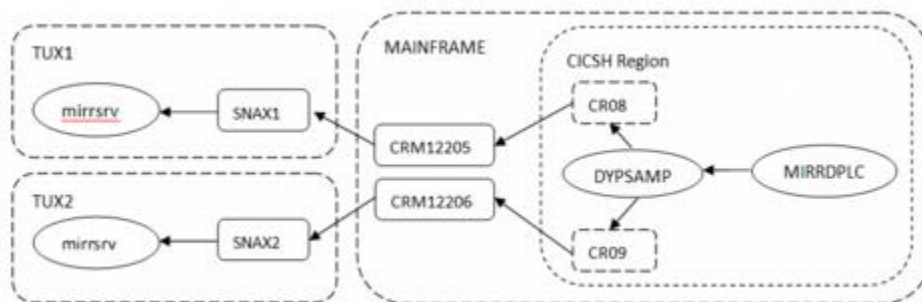
Unlike the single connection and multiple links, in this scenario, each CRM accepts two connections from GWSNAX in two separate Tuxedo domains. One domain serves as a master, and the other serves as a slave.

CRM supports both multiple connections and multiple links simultaneously.

 **Note:**

- Multiple connections is only supported by CRM 12cR1 or later.
- GWSNAXs connect to one CRM must be in different Tuxedo domains.

Figure 10-4 Multiple Connections and Multiple Links



The configurations for multiple connections and multiple links are almost same as that of the single connection scenario and won't be explained again. You can refer to the preceding configuration steps in [Single Connection and Multiple Links](#). There are just some slight differences in the configuration of variables `IPCKEY`, `APPDIR`, `TUXCONFIG`, `TLOGDEVICE` in `UBBCONFIG`, and `DMTLOGDEV` in `DMCONFIG`.

11

ART CICS Calls Transaction in Z/CICS

In certain circumstances, some transactions have been migrated to ART/CICS on Linux, while some still reside on z/OS. This solution aims to build a bridge, through which applications in ART/CICS can invoke transactions on z/OS.

You can call an online transaction, referred to as main transaction hereinafter, provided by ART/CICS from 3270 terminal or a Tuxedo client through GWSNAX. To make the main transaction call a sub-transaction on z/OS, you need to follow these steps, which will be elaborated later:

- [Step 1: Configuring ART CICS on Open System](#)
- [Step 2: Defining VTAM Resources](#)
- [Step 3: Configuring and Deploying TMA SNA](#)
- [Step 4: Start CRM on Mainframe](#)

This chapter contains the following topics:

- [Step 1: Configuring ART CICS on Open System](#)
- [Step 2: Defining VTAM Resources](#)
- [Step 3: Configuring and Deploying TMA SNA](#)
- [Step 4: Start CRM on Mainframe](#)

11.1 Step 1: Configuring ART CICS on Open System

To configure ART CICS on open system, do the following:

- In the `transactions.desc` file at `$KIXCONFIG`, add your ART CICS client transaction:

```
BUFC;DTPAPFT;dpl caller <sysid>_<program>;BUFCLT
```

- In `programs.desc`, add the client and server program:

```
BUFCLT;DTPAPFT;DPL client;COBOL; ;ENABLED
```

```
BUFSRV;DTPAPBK;DPL server;COBOL; ;ENABLED;KIXB
```

For more information, see [CICS Runtime Configuration Files](#).

11.2 Step 2: Defining VTAM Resources

1. Define LU in VTAM configuration:
The LU configuration for the remote CICS region should resemble in the following Listing and be located in `SYS1.VTAMLIST` by default. It is shown in bold typeface where the LU name is used for this remote CICS region.

Listing LU Configuration for Remote CICS Region

```

CRMMN3          VBUILD TYPE=APPL
**LUNAME**      APPL ACBNAME=**LUNAME**,
                APPC=YES,
                SYNCLVL=SYNCPT,
                AUTH=(ACQ,PASS),
                PARSESS=YES
*example ACBNAME=( CRMLU80 )

```

Note:

All entries prefixed and terminated with ** are placeholders for you to fill in with relevant parameters, or take a note of the subsequent configuration entries. It is applicable to all of the subsequent cases and won't be noted again.

2. Check or define the connection in the remote CICS region:
The purpose is to add LU to the connection of the CICS region. The value of NETNAME field should be consistent with the LU defined on system A in [Step 1: Configuring ART CICS on Open System](#).

Note:

When the same placeholder is identified in multiple entries, please use consistent values, for example, **GROUPNAME**, as in the following listing.

Listing Define LU Connection for Remote CICS Region

```

ADD GROUP(**GROUPNAME**) LIST(**LISTNAME**)

DEFINE CONNECTION(**CONNNAME**)  GROUP(**GROUPNAME**)
    DE(BEA TMA SNA EXAMPLE RDO CONNECTION)
    ACCESSMETHOD(VTAM)           PROTOCOL(APPC)
    NETNAME(**LUNAME**)
    ATTACHSEC(LOCAL)             AUTOCONNECT(NO)
*example CONNECTION(BEALST)
*example CONNECTION(BEA)
*example GROUP(BEACONN)
*example NETNAME(CRMLU80)

DEFINE SESSION(**SESSIONNAME**)  GROUP(**GROUPNAME**)
    CONNECTION(**CONNNAME**)
    DE(SESSION EXAMPLE)
    PROTOCOL(APPC)               AUTOCONNECT(YES)
    MODENAME(**SNANAME**)        MAXIMUM(**SESSNBR**,**WINNER**)
*example SESSION(BEATEST)
*example CONNECTION(BEA)
*example GROUP(BEACONN), keep it same as group name in DEFINE
CONNECTION

```

```
statement
*example MODENAME(SMSNA100)      MAXIMUM(10,5)
```

3. Activate defined LU resource:
In SDSF, enter the following commands:
 - a. Enter LOG in SDSF.
 - b. Enter /V net,id>**Major Node Name**,act,scope=all to activate all LUs defined under this major node. Example below uses crmmn3 as the node name.
 - c. Check result, as shown below, the screen shows crmmn3 active

11.3 Step 3: Configuring and Deploying TMA SNA

1. Install Oracle TMA for SNA Gateway on open system and Oracle TMA for SNA CRM on z/OS. For more information, refer to [Oracle Tuxedo Mainframe Adapter for SNA Overview](#).
2. Add one line SNAX:::: to your \$TUXDIR/udataobj/DMTYPE.
3. In Tuxedo UBBCONFIG, configure the following:
 - Add two GROUP entries and three SERVER entries as shown in the following Listing.
Listing UBBCONFIG Configuration

```
*GROUPS

DMGRP      GRPNO>**NN**      LMID>**LMNAME**
GWGRP1     GRPNO>**MM**      LMID>**LMNAME**

*SERVERS
DMADM      SRVID>**nnn**      SRVGRP=DMGRP
GWADM      SRVGRP=GWGRP1   SRVID>**nnn**
           REPLYQ=N  RESTART=Y
GWSNAX     SRVGRP=GWGRP1   SRVID>**nnn**
           RQADDR="**SNADOM**" REPLYQ=N  RESTART=Y
           CLOPT="-A -- -b"
```

 **Note:**

- GRPNO parameters are numeric and must be unique for all groups defined in ubbconfig. LMNAME parameter for the LMID must be one of the LMIDS defined in the *MACHINES section of the ubbconfig. In MP configuration with multiple LMIDS, it is recommended to define multiple groups — one per LMID — for the GWSNAX servers to be replicated across all machines in the cluster.
- **nnn** is the Server ID number that must be unique for each server in the group. RESTART=Y is recommended; -b in CLOPT is recommended if there is a router or firewall between GWSNAX and CRM on z/OS periodically breaking the connection. This option activates a heartbeat mode that can help maintain the connectivity.

After updating the ubbconfig, ensure:

- * the Tuxedo domain is not running (otherwise run tmshutdown), and run tmloadcf to compile the updated ubbconfig;
- * BDMCONFIG environment variable points to the latest compiled bdmconfig file;
- * ULOG establishes connections with the CRM after running tmboot to bring up Tuxedo domain with GWSNAX;
- * logs on the CRM side are able to establish connections with the remote CICS region.

For more information, see [Configuring and Starting the System](#).

- (Optional) Enable service level security.

Listing Enabling service level security in UBBCONFIG

```
*RESOURCES
SECURITY          USER_AUTH
*SERVERS
AUTHSVR           SRVGRP=GROUP0 SRVID=2 CLOPT="-A"
```

- Add ARTTCPL ARTADM ARTCNX ARTTSQ ARTSTRN ARTDPL to the *SERVERS section.

4. Configure DMCONFIG:

Tuxedo dmconfig file is used to define the CRM parameters and services accessed through the TMA SNA connection. Some of these parameters are used locally and others are sent by GWSNAX to CRM on z/OS when the connection is established to dynamically configure the connection attributes. Some of the configuration parameters we have specified are used below in the DM_SNASTACKS, DM_SNALINKS, and DM_REMOTE_SERVICES sections of the dmconfig.

If you are building a new dmconfig, the following listing may need to be adjusted to your mainframe connection. If it already exists for the same CRM connection, you only need to merge the relevant entries from the DM_SNASTACKS, DM_SNALINKS, and

DM_REMOTE_SERVICES sections into the existing configuration. It is shown in bold typeface where you need to change according to your environment in the following listing:

Listing Sample DMCONFIG Configuration

```
*DM_LOCAL_DOMAINS
simpsnad
    GWGRP=GWGRP1 // same as GWSNAX group in ubbconfig
    TYPE=SNAX
    DOMAINID="simpsnad"
    DMTLOGDEV=**tuxedo/path/to/DMTLOG** //GWSNAX domain TLOG file
name
*DM_REMOTE_DOMAINS
MVSDOM TYPE=SNAX
DOMAINID="MVSDOM"
*DM_SNACRM
simpcrm SNACRMADDR="//host.name:NNNN" //host:port of CRM
    NWDEVICE="/dev/tcp"
    LDOM="simpsnad"
*DM_SNASTACKS
stack1
    SNACRM="simpcrm"
    STACKTYPE="VT210"
    LOCALLU="**LUNAME**" // NETNAME in VTAM CONNECTION
definition
    LTPNAME="*"
    STACKPARMS="host.name" //same as CRM host name
*DM_SNALINKS
link1 STACKREF="stack1" //Reference to DM_SNASTACKS entry name
    RDOM="MVSDOM" //Reference to DOMAINID in
DM_REMOTE_DOMAINS
    LSYSID="OT12" //VTAM connection name
    RSYSID="**CICX**"
    RLUNAME="**CICSAPPL**" //CICS APPLID of CICS region
    MODENAME="**SNANAME**" //MODE NAME in SESSION definition.
    SECURITY="LOCAL" //if you want to use security features,
refer
to documentation
    STARTTYPE="COLD"
    MAXSESS=20
    MINWIN=20
    MAXSYNCLVL=2
    *DM_LOCAL_SERVICES
    *DM_REMOTE_SERVICES
    SVSN **PROGY** AUTOTRAN=N
    LDOM="simpsnad" //Reference to DOMAINID in
DM_LOCAL_DOMAINS
    RDOM=MVSDOM //Reference to DOMAINID in
DM_REMOTE_DOMAINS
    CONV=N
    RNAME="**PROGY**" //Reference to program name defined in
remote
CICS region
```

5. Use tmloadcf to load UBBCONFIG.
The application password must match that of ARTCICS.

6. Use `dmloadcf` to load `DMCONFIG`.
7. Add Tuxedo user using Tuxedo-provided tools:

```
tpgrpadd tuxgrp tpusradd -g tuxgrp tuxusr
```

Replace `tuxgrp` and `tuxusr` with your own and three files — `tpgrp`, `tpusr`, and `tpacl` — will be generated.
8. Add user mapping in `dmadmin`:

```
addusr -d SNAXDOM -R MVSDOM -u "mfusr" -w
addumap -d SNAXDOM -R MVSDOM -p "tuxusr" -u "mfusr"
```
9. If the service level security is enabled, logon CICS with `tuxusr/password`.

11.4 Step 4: Start CRM on Mainframe

Use the following command to start CRM on mainframe:

```
CRM [ -t 0|1|2|3 ] [-p<nbr>] [-s] [-n <type>:#:#] [-u <keyfile>] [-S
<SSL configuration file>] [-V] [-P [interval]] [-R [interval]] [-F
[trcsz]] <addr> <group>
```

You can use JCL to start CRM as shown in the following listing:

Listing Using JCL to Start CRM

```
// 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
```

For more information, see [Command Reference Pages](#).

After all preceding steps performed, boot the ARTIMS domain and make a call to the online transaction to invoke the sub-transaction on z/OS.

ART IMS Calls Transaction in Z/IMS

In some circumstances, the main transaction has been migrated to ART/IMS on Linux, while some sub-transactions still reside on z/OS, as they are shared by other main transactions residing on z/OS. This solution aims to enable applications in ART/IMS to invoke sub-transactions on z/OS without code changes, e.g. through Alternate PCB, referred to as ALTPCB hereinafter.

You can call an online transaction, referred to as main transaction hereinafter, provided by ARTIMS from 3270 terminal or a Tuxedo client through ARTIGW. To make the main transaction call a sub-transaction on z/OS, you need to follow these steps, which will be elaborated later:

- [Step 1: Configuring Sub-transactions on Z/OS](#)
- [Step 2: Configuring and Deploying TMA SNA for IMS](#)
- [Step 3: Generating Buffer Converter Program for Sub-transactions](#)

This chapter contains the following topics:

- [Step 1: Configuring Sub-transactions on Z/OS](#)
- [Step 2: Configuring and Deploying TMA SNA for IMS](#)
- [Step 3: Generating Buffer Converter Program for Sub-transactions](#)

12.1 Step 1: Configuring Sub-transactions on Z/OS

You can configure sub-transactions on z/OS in the `zostrans.desc` file at `$ART_IMS_CONFIG`, all of which must only be called sub-transaction. For instance, in the following listing:

Listing Sample Sub-transaction Configuration

```
[zostran]
NAME=SUBTRAN1
CONV=TMAA2Z
[zostran]
NAME=SUBTRAN2
```

`NAME` is the z/OS transaction name. `CONV` is the input message's buffer converter program, which can be generated by the ART Workbench based on a copybook describing the input message/segment structure of the sub-transaction.

Note:

`CONV` handles the data conversion before calling sub-transactions on z/OS. If it is not specified, ART/IMS only does ASCII to EBCDIC conversion by default, for example, treating all bytes in the input message as characters

12.2 Step 2: Configuring and Deploying TMA SNA for IMS

1. Install the Oracle Tuxedo Mainframe Adapter for SNA. For more information, see [Oracle Tuxedo Mainframe Adapter for SNA Overview](#)
2. Add one line `GWSNAX:::` to your `$TUXDIR/udataobj/DMTYPE`.
3. Establish 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. For more information, see [Preparing Mainframe Configurations for CRM Requirements](#)
4. Configure the Oracle TMA SNA for IMS:
 - a. Configure the `UBBCONFIG` file. The following listing shows an example.
Listing Example of UBBCONFIG Configuration

```
*RESOURCES
IPCKEY                40036
DOMAINID              snax
MASTER                SITE1
MAXACCESSERS          400
MAXSERVERS             80
MAXSERVICES           80
MAXCONV               120
MODEL                 SHM
LDBAL                 N
SCANUNIT              5
BLOCKTIME             4
DBBLWAIT              1
*MACHINES
DEFAULT:
"HOSTNAME"
    LMID=SITE1
    TUXDIR="TUXDIR"
    APPDIR="APPDIR"
    TUXCONFIG="APPDIR/tuxconfig"
    TLOGDEVICE="APPDIR/TLOG"
*GROUPS
ADMGRP
    LMID=SITE1 GRPNO=1
CRM12200
    LMID=SITE1 GRPNO=10
APPGRP
    LMID=SITE1 GRPNO=100 TMSCOUNT=2 TMSNAME=TMS
*SERVERS
DEFAULT: CLOPT="-A"
DMADM
    SRVGRP=ADMGRP SRVID=1
GWADM
    SRVGRP=CRM12200 SRVID=11
GWSNAX
    SRVGRP=CRM12200 SRVID=12
    RQADDR="GWSNAX"
```

```

REPLYQ=N RESTART=N GRACE=0
CLOPT="-A -- -t 1"
*SERVICES

```

b. Configure the `DMCONFIG` file. The following listing shows an example.

If you are building a new `dmconfig`, the following listing may need to be adjusted to your mainframe connection. If it already exists for the same CRM connection, you only need to merge the relevant entries from the `DM_SNASTACKS`, `DM_SNALINKS`, and `DM_REMOTE_SERVICES` sections into the existing configuration. It is shown in bold typeface where you need to change according to your environment in the following listing :

Listing Example of DMCONFIG Configuration

```

*DM_LOCAL_DOMAINS
SNAXDOM
    GWGRP=CRM12200 // Same as GWSNAX group name
    TYPE=SNAX
    DOMAINID="SNAXDOM"
    DMTLOGDEV="APPDIR/DMLOG"
*DM_REMOTE_DOMAINS
MVSDOM
    TYPE=SNAX
    DOMAINID="MVSDOM"
*DM_SNACRM
SIMPCR
    SNACRMADDR="//hostname:port" // The port must be same as
listening
port of CRM
    NWDEVICE="/dev/tcp"
    LDOM="SNAXDOM"
*DM_SNASTACKS
SIMPSTK
    SNACRM="SIMPCR"
    STACKTYPE="VT210" // Do not change if CRM is deployed on
Mainframe
    LOCALLU="CRMLU74" //LU name defined in VTAM
    LTPNAME="*"
    STACKPARMS="wasa"
*DM_SNALINKS
SIMPLK
    STACKREF="SIMPSTK"
    RDOM="MVSDOM"
    LSYSID="CR63" // VTAM Connection name for this link
    RSYSID="IMSD"
    RLUNAME="IMSDAPPC" // VTAM netname
    MODENAME="SMSNA100"
    SECURITY="LOCAL"
    STARTTYPE="COLD"
    MAXSESS=8
    MINWIN=4
    MAXSYNCLVL=2
*DM_LOCAL_SERVICES
*DM_REMOTE_SERVICES
TSTNCON1

```

```

        LDOM="SNAXDOM" //Reference to DOMAINID in
DM_LOCAL_DOMAINS
        RDOM="MVSDOM" //Reference to DOMAINID in
DM_REMOTE_DOMAINS
        AUTOTRAN=N
        CONV=N
        RNAME="TSTNCON1" //Reference to program name defined in
remote CICS region
        INBUFTYPE="CARRAY"
        OUTBUFTYPE="CARRAY"
        FUNCTION="APPC"
*DM_ROUTING

```

5. Perform a `tmloadcf` and `dmloadcf` to load the `UBBCONFIG` and the `DMCONFIG` files.
6. Use the following command to start CRM on mainframe: `CRM [-t 0|1|2|3] [-p<nbr>] [-s] [-n<type>:#:#] [-u<keyfile>] [-V] [-P [interval]] [-R [interval]] [-F [trcsz]]<addr><group>`
The port number of `addr` must be same as that of `SNACRMADDR` configured in `DMCONFIG`; the group name must be same as that of `GWSNAX` configured in `UBBCONFIG`, which is `CRM12200` in the example of `UBBCONFIG`.
For more information, see [CRM](#).
7. Start the `GWSNAX tmbboot` to start all ATMI servers including `GWSNAX`.

12.3 Step 3: Generating Buffer Converter Program for Sub-transactions

To enable this functionality in ART Workbench, you need to build a workbench project to convert files. Then, based on the project, you need to follow these steps to generate the buffer converter program:

1. Configure `enable-buffer-converter` in the `system.desc` file (special option).
2. Put the input copybook file at the `source/COPY` directory.
3. Add the following lines to the `param/file/Datamap-xxx.re` file:

```
file A.FILE1
organization Sequential
```

4. Add the following lines to the `param/file/mapper-xxx.re` file:

```
file A.FILE1 transferred
include "COPY/ODCSF0.cpy"
map record RECORD defined in "COPY/ODCSF0.cpy"
source record RECORD defined in "COPY/ODCSF0.cpy"
logical name ODCSF0Q
converter name ODCSF0Q
```

5. Run the Workbench file converter project again:
Two output files are generated and only the first one is the needed converter program:

```
trf/reload/file/STFILEORA/ODCSF0Q1.cb1(ASCII to EBCDIC)
```

```
trf/reload/file/STFILEORA/ODCSF0Q2.cb1 (EBCDIR to ASCII)
```

For more information, see [Oracle Tuxedo Rehosting Workbench Users Guide](#).

After all preceding steps performed, configure and setup ARTIMS with TMA gateway environment. Boot the ARTIMS domain and make a call to the online transaction to invoke the sub-transaction on z/OS.

A

Administrative Command Reference Pages

See Command Reference Pages in the [Oracle Tuxedo Mainframe Adapter for SNA CRM Administration Guide](#) for the following CRM administration commands:

- CRM
- CRMLOGS
- crmlkoff
- crmlkon

This section covers the following reference pages for administrative commands, formerly called manual pages:

- [addumap](#)
- [addusr](#)
- [delumap](#)
- [delusr](#)
- [DMADM](#)
- [dmadmin](#)
- [dmconfig](#)
- [dmloadcf](#)
- [dmunloadcf](#)
- [GWADM](#)
- [GWSNAX](#)
- [modusr](#)
- [tadmin](#)

A.1 addumap

Adds a local-to-remote mapping for a local/remote domain pair.

- [Synopsis](#)
- [Description](#)
- [Portability](#)
- [Diagnostics](#)
- [Example](#)

A.1.1 Synopsis

```
addumap -d <local domain ID> -R <remote domain ID>  
-p <local principal name> -u <remote username>
```

A.1.2 Description

addumap can only be executed as a subcommand of `dmadmin(1)`. The purpose of this page is to describe options for the subcommand and to show examples.

The subcommand allows the administrator to add local-to-remote user mappings for a local/remote domain pair.

Mappings are defined to be inbound, outbound or both when the application is using SNA-type gateways and `SECURITY` is set to `USER_AUTH`, `ACL`, or `MANDATORY ACL` in the `ubconfig` file and `SECURITY` is set to `DM_PW` or `USER_PW` in the `DMCONFIG` file.

The following options are available:

-d <local domain ID>

This is the name of the local domain gateway with which the ids and passwords are associated. This is the same ID as the one used when creating the domain definitions either in the `DMCONFIG` file or through the Graphical Administrative Interface.

-R <remote domain ID>

This is the name of the remote domain gateway with which the ids and passwords are associated. This is the same ID as the one used when creating the domain definitions either in the `DMCONFIG` file or through the Graphical Administrative Interface.

-p <local principal>

The user identification number. The *local principal* must be defined in the ACL user file and must be unique within the list of existing identifiers for the application.

-u <remote username>

The remote user name as defined in the ACL security application (for example, RACF) of the remote domain.

Before running this subcommand the application must be configured using either the Graphical Administrative Interface or `tmloadcf(1)` and `dmloadcf(1)`. `dmadmin addumap` may be run on any active node.

A.1.3 Portability

This subcommand is available on the latest version of Tuxedo, as documented for this release of Oracle Tuxedo Mainframe Adapter for SNA.

A.1.4 Diagnostics

The `dmadmin addumap` subcommand exits with a return code of 0 upon successful completion.

A.1.5 Example

```
addumap -d ldom -R cdom -p tuxusr -u CICSUSR
/*maps principal tuxusr with
remote user cicsusr */
```



See Also:

dmadmin(1), delumap(5)

A.2 addusr

Adds a user to the remote domain user and password file.

- [Synopsis](#)
- [Description](#)
- [Portability](#)
- [Diagnostics](#)
- [Examples](#)

A.2.1 Synopsis

```
addusr -d <local domain ID> -R <remote domain ID> -u <remote username> [-w ]
```

A.2.2 Description

`addusr` can only be executed as a subcommand of `dmadmin(1)`. The purpose of this page is to describe options for the subcommand and to show an example.

The subcommand allows the administrator to add remote user names and passwords to the remote domain remote user and password table. If `-w` is not specified, the user is prompted for a password.

The table entries created are used for passing remote user names and passwords to remote SNA domains when the application is using SNA-type gateways and `SECURITY` is set to `USER_AUTH`, `ACL`, or `MANDATORY ACL` in the `ubbconfig` file and `SECURITY` is set to `DM_PW` or `USER_PW` in the `DMCONFIG` file.

The following options are available:

-d <local domain ID>

This is the name of the local domain gateway with which the ids and passwords are associated. This is the same ID as the one used when creating the domain definitions either in the `DMCONFIG` file or through the Graphical Administrative Interface.

-R *<remote domain ID >*

This is the name of the remote domain gateway with which the ids and passwords are associated. This is the same ID as the one used when creating the domain definitions either in the `DMCONFIG` file or through the Graphical Administrative Interface.

-u *<remote username>*

The remote user name to be added.

-w

Do not prompt for password.

Before running this subcommand the application must be configured using either the Graphical Administrative Interface or `tmloadcf(1)` and `dmloadcf(1)`. `dmadmin addusr` may be run on any active node.

A.2.3 Portability

This subcommand is available on the latest version of Tuxedo, as documented for this release of Oracle Tuxedo Mainframe Adapter for SNA.

A.2.4 Diagnostics

The `dmadmin addusr` subcommand exits with a return code of 0 upon successful completion.

A.2.5 Examples

```
addusr -d tux -R cics -u CICSUSR      /*adds remote user CICSUSR to
                                     cics domain's user and
                                     password file. The
                                     administrator is prompted for
                                     a password*/
```



See Also:

`delusr(5)`, `modusr(5)`

A.3 delumap

Deletes a local-to-remote mapping for a local/remote domain pair.

- [Synopsis](#)
- [Description](#)
- [Portability](#)
- [Diagnostics](#)
- [Example](#)

A.3.1 Synopsis

```
delumap -d <local domain ID> -R <remote domain ID>  
-p <local principal name> -u <remote username>
```

A.3.2 Description

delumap can only be executed as a subcommand of dmadmin(1). The purpose of this page is to describe options for the subcommand and to show examples.

The subcommand allows the administrator to delete local-to-remote user mappings for a local/remote domain pair.

Mappings are defined to be inbound, outbound or both when the application is using SNA-type gateways and SECURITY is set to USER_AUTH, ACL, or MANDATORY ACL in the ubbconfig file and SECURITY is set to DM_PW or USER_PW in the DMCONFIG file.

The following options are available:

-d <local domain ID>

This is the name of the local domain gateway with which the ids and passwords are associated. This is the same ID as the one used when creating the domain definitions either in the DMCONFIG file or through the Graphical Administrative Interface.

-R <remote domain ID>

This is the name of the remote domain gateway with which the ids and passwords are associated. This is the same ID as the one used when creating the domain definitions either in the DMCONFIG file or through the Graphical Administrative Interface.

-p <local principal>

The user identification number. The *local principal* must be defined in the ACL user file and must be unique within the list of existing identifiers for the application.

-u <remote username>

The remote user name as defined in the ACL security application (for example, RACF) of the remote domain. Space is a valid remote username.

Before running this subcommand the application must be configured using either the Graphical Administrative Interface or tmloadcf(1) and dmloadcf(1). dmadmin delumap may be run on any active node.

A.3.3 Portability

This subcommand is available on the latest version of Tuxedo, as documented for this release of Oracle Tuxedo Mainframe Adapter for SNA.

A.3.4 Diagnostics

The dmadmin delumap subcommand exits with a return code of 0 upon successful completion.

A.3.5 Example

```
delumap -d ldom -R cics -p tuxusr -u CICSUSR
/*deletes the mapping of principal
tuxusr with remote user cicsusr */
```



See Also:

[dmadmin\(1\)](#), [addumap\(5\)](#)

A.4 delusr

Deletes a user from the remote domain user and password file.

- [Synopsis](#)
- [Description](#)
- [Portability](#)
- [Diagnostics](#)
- [Examples](#)

A.4.1 Synopsis

```
delusr -d <local domain> -R <remote domain> -u <remote username>
```

A.4.2 Description

`delusr` can only be executed as a subcommand of `dmadmin(1)`. The purpose of this page is to describe options for the subcommand and to show an example.

The subcommand allows the administrator to remove remote user names and passwords from the remote domain remote user and password table.

Once the entries are deleted they can no longer be used for mapping remote user names and passwords to local user names and passwords when the application is using SNA-type gateways and `SECURITY` is set to `USER_AUTH`, `ACL`, or `MANDATORY ACL` in the `ubbconfig` file and `SECURITY` is set to `DM_USER_PW` in the `DMCONFIG` file.

The following options are available:

-d <local domain ID>

This is the name of the local domain gateway with which the ids and passwords are associated. This is the same ID as the one used when creating the domain definitions either in the `DMCONFIG` file or through the Graphical Administrative Interface.

-R <remote domain ID>

This is the name of the remote domain gateway with which the ids and passwords are associated. This is the same ID as the one used when creating the domain definitions either in the `DMCONFIG` file or through the Graphical Administrative Interface.

-u <remote username>

The remote user name to be deleted.

Before running this subcommand the application must be configured using either the Graphical Administrative Interface or `tmloadcf(1)` and `dmloadcf(1)`. `dmadmin delusr` may be run on any active node.

A.4.3 Portability

This subcommand is available on the latest version of Tuxedo, as documented for this release of Oracle Tuxedo Mainframe Adapter for SNA.

A.4.4 Diagnostics

The `dmadmin delusr` subcommand exits with a return code of 0 upon successful completion.

A.4.5 Examples

```
delusr -d tux -R cics -u CICSUSR      /*deletes remote user CICSUSR to
                                     cics domain users. The
                                     administrator is prompted for a
                                     password*/
```



See Also:

`addusr(5)`, `modusr(5)`

A.5 DMADM

/Domain administrative server.

- [Synopsis](#)
- [Description](#)
- [Portability](#)
- [Examples](#)

A.5.1 Synopsis

```
DMADM SRVGRP = "identifier"
        SRVID = "number"
        REPLYQ = "N"
```

A.5.2 Description

The /DOMAIN administrative server (DMADM) is a Tuxedo-supplied server that provides run-time access to the binary domain configuration file (BDMCONFIG file). When DMADM is booted, the BDMCONFIG environment variable must be set to the pathname of the file containing the binary version of the DMCONFIG file.

DMADM is described in the SERVERS section of the UBBCONFIG file as a server running within a group, e.g., DMADMGRP. There must be only one instance of the DMADM running in this group and it must not have a reply queue (REPLYQ must be set to "N").

The following server parameters can also be specified for the DMADM server in the SERVERS section: SEQUENCE, ENVFILE, MAXGEN, GRACE, RESTART, RQPERM and SYSTEM_ACCESS.

A.5.3 Portability

DMADM is supported as a Tuxedo-supplied server on UNIX System and Windows NT operating systems.

A.5.4 Examples

The following example illustrates the definition of the administrative server and a gateway group in the UBBCONFIG file.

```
#
*GROUPS
DMADMGRP  LMID=mach1 GRPNO=1
gwgrp     LMID=mach1 GRPNO=2
#
*SERVERS
DMADM SRVGRP="DMADMGRP" SRVID=1001 REPLYQ=N RESTART=Y GRACE=0
GWADM SRVGRP="gwgrp" SRVID=1002 REPLYQ=N RESTART=Y GRACE=0
GWSNAX SRVGRP="gwgrp" SRVID=1003 RQADDR="gwgrp" REPLYQ=N
RESTART=N MIN=1 MAX=1
```



Note:

dmadmin(1), tmboot(1), dmconfig(5), GWADM(5), servopts(5),
ubbconfig(5) *Using the ORACLE Tuxedo Domains Component*

A.6 dmadmin

Tuxedo System/T Domain Administration Command Interpreter.

- [Synopsis](#)
- [Description](#)

- Administration Mode Commands
- Configuration Mode Commands
- Configuration Input Format
- Configuration Limitations
- Restrictions for Configuration Field Identifiers/Updates
- Configuring the DM_LOCAL_DOMAINS Section
- Configuring the DM_REMOTE_DOMAINS Section
- Configuring the DM_TDOMAIN Section
- Configuring the DM_OSITP Section
- Configuring the DM_LOCAL_SERVICES Section
- Configuring the DM_REMOTE_SERVICES Section
- Configuring the DM_ROUTING Section
- Configuring the DM_ACCESS_CONTROL Section
- Configuring the DM_PASSWORDS Section
- Diagnostics in Configuration Mode
- Configuration Example
- Security
- Environment Variables
- General Diagnostics
- Interoperability
- Portability

A.6.1 Synopsis

```
dmadmin [-c]
```

A.6.2 Description

The `dmadmin` interactive command interpreter is used for the administration of domain gateway groups defined for a particular Tuxedo System/T application. The interpreter can operate in two modes: administration mode and configuration mode.

The `dmadmin` command interpreter enters *administration* mode when called with no parameters. This is the default. In this mode, `dmadmin` can be run on any active node (excluding workstations) within an active application. Application administrators can use this mode to obtain or change parameters on any active domain gateway group. Application administrators may also use this mode to create, destroy, or re-initialize the `DMTLOG` for a particular local domain. In this case, the domain gateway group associated with that local domain must not be active, and `dmadmin` must be run on the machine assigned to the corresponding gateway group.

The `dmadmin` command interpreter enters *configuration* mode when it is invoked with the `-c` option or when the `config` subcommand is invoked. Application administrators can use this

mode to update or add new configuration information to the binary version of the domain configuration file (`BDMCONFIG`).

The `dmadmin` command interpreter requires the use of the `DOMAIN` administrative server (`DMADM`) for the administration of the `BDMCONFIG` file and the gateway administrative server (`GWADM`) for the re-configuration of active `DOMAIN` gateway groups (there is one `GWADM` per gateway group).

A.6.3 Administration Mode Commands

Once `dmadmin` has been invoked, commands may be entered at the prompt (“>”) according to the following syntax:

```
command [arguments]
```

Several commonly occurring arguments can be given default values using the default command. Commands that accept parameters set using the default command. Check default to see if a value has been set. If no value is set, an error message is returned.

Once set, a default value remains in effect until the session is ended, unless changed by another default command. Defaults may be overridden by entering an explicit value on the command line, or reset by entering the value “*”. The effect of an override lasts for a single instance of the command.

Output from `dmadmin` commands is paginated according to the pagination command in use (see the `paginate` subcommand below).

Commands may be entered either by their full name or their abbreviation (shown in parentheses) followed by any appropriate arguments. Arguments appearing in square brackets, [], are optional; those in curly braces, {}, indicate a selection from mutually exclusive options. Note that for many commands `local_domain_name` is a required argument, but commands can be set with the default command.

The following commands are available in administration mode:

addumap [options]

Add local user mappings to remote user mappings for a local/remote domain pair. Mappings are defined to be inbound, outbound or both. See the `addumap(5)` reference page for an explanation of the available options and for examples.

addusr (addu) [options]

Add remote user names and passwords to the remote user and password tables of a remote domain. See the `addusr(5)` reference page for an explanation of the available options and for examples.

advertise (adv) -d local_domain_name[{ -all | service}]

Advertise all remote services provided by the named local domain or the specified remote service.

audit (audit) -d local_domain_name[{off | on}]

Activate (on) or deactivate (off) the audit trace for the named local domain. If no option is given, then the current setting will be toggled between the values `on` and `off`, and the new setting will be printed. The initial setting is `off`.

chbktime (chbt) -d local_domain_name -t bktime
Change the blocking timeout for a particular local domain.

config (config)
Enter configuration mode. Commands issued in this mode follow the conventions defined in the section “Configuration Mode Commands” (see below).

crdmlog (crdlg) -d local_domain_name
Create the domain transaction log for the named local domain on the current machine (that is, the machine where `dmadmin` is running). The command uses the parameters specified in the `DMCONFIG` file. This command fails if the named local domain is active on the current machine or if the log already exists.

default (d) [-d local_domain_name]
Set the corresponding argument to be the default local domain. Defaults may be reset by specifying “*” as an argument.

If the `default` command is entered with no arguments, the current default values are printed.

delumap [options]
Delete local to remote user mappings for a local/remote domain pair. See the `delumap(5)` reference page for an explanation of the available options and for examples.

delusr (delu) [options]
Delete remote user names and passwords from the remote user and password tables of a remote domain. See the `delusr(5)` reference page for an explanation of the available options and for examples.

dsdmlog (dsdlg) -d local_domain_name [-y]
Destroy the domain transaction log for the named local domain on the current machine (that is, the machine where `dmadmin` is running). An error is returned if a `DMTLOG` is not defined for this local domain, if the local domain is active, or if outstanding transaction records exist in the log. The term outstanding transactions means that a global transaction has been committed but an end-of-transaction has not yet been written. This command prompts for confirmation before proceeding unless the `-y` option is specified. `dsdmlog` is not supported for SNA-type gateways.

echo (e) [{off | on}]
Echo input command lines when set to `on`. If no option is given, then the current setting is toggled, and the new setting is printed. The initial setting is `off`.

forgettrans (ft) -d local_domain_name [-t tran_id]
Forget one or all heuristic log records for the named local domain. If the transaction identifier `tran_id` is specified, then only the heuristic log record for that transaction will be forgotten. The transaction identifier `tran_id` can be obtained from the `printtrans` command or from the `ULOG` file. `forgettrans` is not supported for SNA-type gateways.

help (h) [command]
Print help messages. If `command` is specified, the abbreviation, arguments, and description for that command are printed. Omitting all arguments causes the syntax of all commands to be displayed.

indmlog (indlg) -d local_domain_name [-y]
Re-initialize the domain transaction log for the named local domain on the current machine (that is, the machine where `dmadmin` is running). An error is returned if a `DMTLOG` is not

defined for this local domain, if the local domain is active, or if outstanding transaction records exist in the log. The term outstanding transactions means that a global transaction has been committed but an end-of-transaction has not yet been written. The command prompts for confirmation before proceeding unless the `-y` option is specified. `indmlog` is not supported for SNA-type gateways.

modusr (modu) [options]

Change remote passwords in the password tables of a remote domain. See the `modusr(5)` reference page for an explanation of the available options and for examples.

paginate (page) [{off | on}]

Paginate output. If no option is given, then the current setting will be toggled, and the new setting is printed. The initial setting is on, unless either standard input or standard output is a non-tty device. Pagination may only be turned on when both standard input and standard output are tty devices. The shell environment variable `PAGER` may be used to override the default command used for paging output. The default paging command is the indigenous one to the native operating system environment, for example, the command `pg` is the default on UNIX System operating environments.

passwd (passwd) [-r] local_domain_name remote_domain_name

Prompts the administrator for new passwords for the specified local and remote domains. The `-r` option specifies that existing passwords and new passwords must be encrypted using a new key generated by the system. The password is truncated after at most eight characters.

printdomain (pd) -d local_domain_name

Print information about the named local domain. Information printed includes connected remote domains, global information shared by the gateway processes, and additional information that is dependent on the domain type instantiation.

printstats (stats) -d local_domain_name

Print statistical and performance information gathered by the named local domain. The information printed is dependent on the domain gateway type.

printtrans (pt) -d local_domain_name

Print transaction information for the named local domain. `printtrans` is not supported for SNA-type gateways.

quit (q)

Terminate the session.

resume (res) -d local_domain_name [{ -all | service}]

Resume processing of the specified service or for all remote services handled by the named local domain.

stats (stats) -d local_domain_name[{ off | on | reset }]

Activate (`on`), deactivate (`off`), or reset (`reset`) statistics gathering for the named local domain. If no option is given, then the current setting will be toggled between the values `on` and `off` , and the new setting will be printed. The initial setting is `off`.

suspend (susp) -d local_domain_name [{ -all | service}]

Suspend one or all remote services for the named local domain.

statsvc (statsvc) -d local_domain_access_point_name -t interval time number
 Activate (on) or deactivate (off) the statistics audit trace for the named local domain access point with remote service call. If the interval time number is greater than 0, it takes effect when the feature is activated. The initial setting is off. When the statistics audit trace is activated, GWADM will not flush the data to audit file if there is no event occurred in the interval time. GWADM only flushes the new data occurred in an interval time to audit log file.

unadvertise (unadv) -d local_domain_name[{ -all | service}]
 Unadvertise one or all remote services for the named local domain.

verbose (v) [{off | on}]
 Produce output in verbose mode. If no option is given, then the current setting will be toggled, and the new setting is printed. The initial setting is off.

! shellcommand
 Escape to shell and execute *shellcommand* .

!!
 Repeat previous shell command.

[text]
 Lines beginning with "#" are comment lines and are ignored.

<CR>
 Repeat the last command.

A.6.4 Configuration Mode Commands

The `dmadmin` command enters configuration mode when executed with the `-c` option or when the `config` subcommand is used. In this mode, `dmadmin` allows run-time updates to the `BDMCONFIG` file. `dmadmin` manages a buffer that contains input field values to be added or retrieved, and displays output field values and status after each operation completes. The user can update the input buffer using any available text editor.

The `dmadmin` command first prompts for the desired section followed by a prompt for the desired operation.

The prompt for the section is as follows:

Sections:

- 1) LOCAL_DOMAINS 2) REMOTE_DOMAINS
 - 3) LOCAL_SERVICES 4) REMOTE_SERVICES
 - 5) ROUTING 6) ACCESS_CONTROL
 - 7) PASSWORDS 8) TDOMAIN
 - 9) OSITP 10) SNA
 - 11) QUIT
- Enter Section [1]:

The number of the default section appears in square brackets at the end of the prompt. You can accept the default by pressing RETURN or ENTER. To select another section enter its number, then press RETURN or ENTER.

dmadmin then prompts for the desired operation.

Operations:

```
1) FIRST 2) NEXT
3) RETRIEVE 4) ADD
5) UPDATE 6) DELETE
7) NEW_SECTION 8) QUIT
Enter Operation [1]:
```

The number of the default operation is printed in square brackets at the end of the prompt. Pressing RETURN or ENTER selects this option. To select another operation enter its number, then press RETURN or ENTER.

The currently supported operations are:

1. FIRST

Retrieve the first record from the specified section. No key fields are needed (they are ignored if in the input buffer).

2. NEXT

Retrieve the next record from the specified section, based on the key fields in the input buffer.

3. RETRIEVE

Retrieve the indicated record from the specified section by key field(s) (see fields description below).

4. ADD

Add the indicated record in the specified section. Any fields not specified (unless required) take their default values as specified in `dmconfig(5)`. The current value for all fields is returned in the output buffer. This operation can only be done by the System/T administrator.

5. UPDATE

Update the record specified in the input buffer in the selected section. Any fields not specified in the input buffer remain unchanged. The current value for all fields is returned in the input buffer. This operation can only be done by the System/T administrator.

6. DELETE

Delete the record specified in the input buffer from the selected section. This operation can only be done by the System/T administrator.

7. NEW SECTION

Clear the input buffer (all fields are deleted). After this operation, dmadmin immediately prompts for the section again.

8. QUIT

Exit the program gracefully (`dmadmin` is terminated). A value of `q` for any prompt also exits the program.

For configuration operations, the effective user identifier must match the System/T administrator user identifier (UID) for the machine on which this program is executed. When a record is updated or added, all default values and validations used by `dmloadcf(1)` are enforced.

`dmadmin` then prompts whether or not to edit the input buffer.

```
Enter editor to add/modify fields [n]?
```

Entering a value of `y` will put the input buffer into a temporary file and execute the text editor. The environment variable `EDITOR` is used to determine which editor to be used; the default is “`ed`”. The input format is in field name/field value pairs and is described in the `CONFIGURATION INPUT FORMAT` section below. The field names associated with each `DMCONFIG` section are listed in tables in the subsections below. The semantics of the fields and associated ranges, default values, restrictions, etc., are described in `dmconfig(5)`. In most cases, the field name is the same as the `KEYWORD` in the `DMCONFIG` file, prefixed with “`TA_`”. When the user completes editing the input buffer, `dmadmin` reads it. If more than one line occurs for a particular field name, the first occurrence is used and other occurrences are ignored. If any errors occur, a syntax error will be printed and `dmadmin` prompts whether or not to correct the problem.

```
Enter editor to correct?
```

If the problem is not corrected (response `n`), then the input buffer will contain no fields. Otherwise, the editor is executed again.

Finally, `dmadmin` asks if the operation must be done.

```
Perform operation [y]?
```

When the operation completes, `dmadmin` prints the return value as in

```
Return value TAOK
```

followed by the output buffer fields. The process then begins again with a prompt for the section. All output buffer fields are available in the input buffer unless the buffer is cleared.

Entering break at any time restarts the interaction at the prompt for the section.

When “`QUIT`” is selected, `dmadmin` prompts for authorization to create a backup ASCII version of the configuration:

```
Unload BDMCONFIG file into ASCII backup [y]?
```

If a backup is selected, `dmadmin` prompts for the file name.

```
Backup filename [DMCONFIG]?
```

On success, `dmadmin` indicates that a backup was created, otherwise an error is printed.

A.6.5 Configuration Input Format

Input packets consist of lines formatted as follows:

```
fldname<tabs>fldval
```

The field name is separated from the field value by one or more tabs (or spaces).

Lengthy field values can be continued on the next line by having the continuation line begin with one or more tabs (which are dropped when read back into `dmadmin`).

Empty lines consisting of a single newline character are ignored.

To enter an unprintable character in the field value or to start a field value with a tab, use a backslash followed by the two-character hexadecimal representation of the desired character (see ASCII(5) in a UNIX reference manual). A space, for example, can be entered in the input data as `\20`. A backslash can be entered using two backslash characters. `dmadmin` recognizes all input in this format, but its greatest usefulness is for non-printing characters.

A.6.6 Configuration Limitations

The following are general limitations of the dynamic domain re-configuration capability:

- Values for key fields (as indicated in the following sections) may not be modified. Key fields can be modified, when the system is down, by reloading the configuration file.
- Dynamic deletions cannot be applied when local domains are active (the corresponding gateway group is running).

A.6.7 Restrictions for Configuration Field Identifiers/Updates

The following sections describe the following information for each `DMCONFIG` section:

- Field identifiers for each `DMCONFIG` field
- Field type of identifier
- Field updates

All applicable field values are returned with the retrieval operations. Fields that are allowed and/or required for adding a record are described in `dmconfig(5)`. Fields indicated below as *key* are key fields that are used to uniquely identify a record within section. These key fields are required to be in the input buffer when updates are done and are not allowed to be updated dynamically. The `Update` column indicates when a field can be updated. The possible values are:

Yes

Can be updated at any time.

NoGW

Cannot be updated dynamically while the gateway group representing the local domain is running.

No

Cannot be updated dynamically while at least one gateway group is running.

A.6.8 Configuring the DM_LOCAL_DOMAINS Section

The following table lists the fields in the DM_LOCAL_DOMAINS section.

Table A-1 DM_LOCAL_DOMAINS SECTION

Field Identifier	Field Type	Update	Notes
TA_LDOM	string	NoGW	Key
TA_AUDITLOG	string	Yes	-
TA_BLOCKTIME	numeric	Yes	-
TA_DOMAINID	string	NoGW	-
TA_DMTLOGDEV	string	NoGW	-
TA_DMTLOGNAME	string	NoGW	-
TA_DMTLOGSIZE	numeric	NoGW	-
TA_GWGRP	string	NoGW	-
TA_MAXDATALEN	numeric	Yes	-
TA_MAXRDOM	numeric	Yes	-
TA_MAXRDTRAN	numeric	NoGW	-
TA_MAXTRAN	numeric	NoGW	-
TA_SECURITY	string	Yes	format: {NONE APP_PW DM_PW}
TA_TYPE	string	NoGW	format: {TDOMAIN OSITP SNA}

A.6.9 Configuring the DM_REMOTE_DOMAINS Section

The following table lists the fields in the DM_REMOTE_DOMAINS section.

Table A-2 DM_REMOTE_DOMAINS SECTION

Field Identifier	Field Type	Update	Notes
TA_RDOM	string	No	key
TA_DOMAINID	string	No	-
TA_TYPE	string	No	format: {TDOMAIN OSITP SNA}
TA_CODEPAGE	string	No	CODEPAGE filename

A.6.10 Configuring the DM_TDOMAIN Section

The DM_TDOMAIN section contains the network addressing parameters required by TDOMAIN type domains. The following lists the fields in this section:

Table A-3 DM_TDOMAIN SECTION

Field Identifier	Field Type	Update	Notes
TA_LDOM or TA_RDOM	string	No/NoGW	key
TA_NWADDR	string	No/NoGW	ASCII format (no embedded NULL characters)

If the domain identifier (TA_LDOM) is a local domain identifier, then the TA_NWADDR field can be updated if the gateway group representing that local domain is not running.

Table A-4 DM_OSITP SECTION

Field Identifier	Field Type	Update	Notes
TA_LDOM or TA_RDOM	string	No/NoGW	key
TA_APT	string	No/NoGW	-
TA_AEQ	string	No/NoGW	-
TA_AEQ	string	No/NoGW	-
TA_ACN	string	No/NoGW	-
TA_APID	string	No/NoGW	-
TA_AEID	string	No/NoGW	-
TA_PROFILE	string	No/NoGW	-

A.6.11 Configuring the DM_OSITP Section

The DM_OSITP section contains the network addressing parameters required by OSITP type domains. The following lists the fields in this section:

If the domain identifier (TA_LDOM) is a local domain identifier, then the other fields in this table can be updated if the gateway group representing that local domain is not running.

A.6.12 Configuring the DM_LOCAL_SERVICES Section

The following table lists the fields in the DM_LOCAL_SERVICES section.

Table A-5 DM_LOCAL_SERVICES SECTION

Field Identifier	Field Type	Update	Notes
TA_SERVICENAME	string	No	key

Table A-5 (Cont.) DM_LOCAL_SERVICES SECTION

Field Identifier	Field Type	Update	Notes
TA_LDOM	string	Yes	-
TA_RNAME	string	Yes	-
TA_ACLNAME	string	Yes	-
TA_BUFTYPE	string	Yes	-
TA_BUFSTYPE	string	Yes	-
TA_OBUFTYPE	string	Yes	-
TA_OBUFSTYPE	string	Yes	-

A.6.13 Configuring the DM_REMOTE_SERVICES Section

The following table lists the fields in the DM_REMOTE_SERVICES section.

Table A-6 DM_REMOTE_SERVICES SECTION

Field Identifier	Field Type	Update	Notes
TA_SERVICENAME	string	No	key
TA_RDOM	string	No	key
TA_LDOM	string	No	key
TA_RNAME	string	Yes	-
TA_CONV	string	NoGW	format: {Y N}
TA_BUFTYPE	string	Yes	-
TA_BUFSTYPE	string	Yes	-
TA_OBUFTYPE	string	Yes	-
TA_OBUFSTYPE	string	Yes	-
TA_ROUTINGNAME	string	Yes	-
TA_TRANTIME	numeric	Yes	-
TA_FUNCTION	string	No	-

A.6.14 Configuring the DM_ROUTING Section

The following table lists the fields in the DM_ROUTING section.

Table A-7 DM_ROUTING SECTION

Field Identifier	Field Type	Update	Notes
TA_ROUTINGNAME	string	No	key
TA_FIELD	string	Yes	-
TA_RANGE	string	Yes	-
TA_BUFTYPE	string	Yes	-
TA_ACLNAME	string	No	key
TA_RDOM	string	Yes	-

A.6.15 Configuring the DM_ACCESS_CONTROL Section

The following table lists the fields in the DM_ACCESS_CONTROL section.

Table A-8 DM_PASSWORDS SECTION

Field Identifier	Field Type	Update	Notes
TA_ACLNAME	string	No	key
TA_RDOM	string	Yes	-

A.6.16 Configuring the DM_PASSWORDS Section

The following table lists the fields in the DM_PASSWORDS section.

Table A-9 DM_PASSWORDS SECTION

Field Identifier	Field Type	Update	Notes
TA_LDOM	string	No	key
TA_RDOM	string	No	key
TA_LPWD	string	Yes	format: {Y N U}
TA_RPWD	string	Yes	format: {Y N U}

The TA_LPWD and TA_RPWD show the existence of a defined password for the local and/or the remote domain. Passwords are not displayed. If an UPDATE operation is selected, the value of the corresponding field must be set to U. The program will then prompt with echo turned off for the corresponding passwords.

A.6.17 Diagnostics in Configuration Mode

dmadmin fails if it cannot allocate an FML typed buffer, if it cannot determine the /etc/passwd entry for the user, or if it cannot reset the environment variables FIELDTBLS or FLDTBLDIR.

The return value printed by dmadmin after each operation completes indicates the status of the requested operation. There are three classes of return values.

The following return values indicate a problem with permissions or a Tuxedo System/T communications error. They indicate that the operation did not complete successfully.

[TAEPERM]

The calling process specified an ADD, UPDATE, or DELETE operation but it is not running as the System/T administrator. Update operations must be run by the administrator (that is, the user specified in the UID attribute of the RESOURCES section of the TUXCONFIG file).

[TAESYSTEM]

A Tuxedo System/T error has occurred. The exact nature of the error is written to `userlog(3)`.

[TAEOS]

An operating system error has occurred.

[TAETIME]

A blocking timeout occurred. The input buffer is not updated so no information is returned for retrieval operations. The status of update operations can be checked by doing a retrieval on the record that was being updated.

The following return values indicate a problem in doing the operation itself and generally are semantic problems with the application data in the input buffer. The string field `TA_STATUS` will be set in the output buffer and will contain short text describing the problem. The string field `TA_BADFLDNAME` will be set to the field name for the field containing the value that caused the problem (assuming the error can be attributed to a single field).

[TAECONFIG]

An error occurred while reading the `BDMCONFIG` file.

[TAEDUPLICATE]

The operation attempted to add a duplicate record.

[TAEINCONSIS]

A field value or set of field values are inconsistently specified.

[TAENOTFOUND]

The record specified for the operation was not found.

[TAENOSPACE]

The operation attempted to do an update but there was not enough space in the `BDMCONFIG` file.

[TAERANGE]

A field value is out of range or is invalid.

[TAERQUIRED]

A field value is required but not present.

[TAESIZE]

A field value for a string field is too long.

[TAEUPDATE]

The operation attempted to do an update that is not allowed.

The following return values indicate that the operation was successful.

[TAOK]

The operation succeeded. No updates were done to the `BDMCONFIG` file.

[TAUPDATED]

The operation succeeded. Updates were made to the `BDMCONFIG` file.

When using `dmunloadcf` to print entries in the configuration, optional field values are not printed if they are not set (for strings) or 0 (for integers). These fields will always appear in the output buffer when using `dmadmin`. In this way, it makes it easier for the administrator to

retrieve an entry and update a field that previously was not set. The entry will have the field name followed by a tab but no field value.

A.6.18 Configuration Example

In the following example, `dadmin` is used to add a new remote domain. For illustration purposes, `ed` is used for the editor.

```
$ EDITOR=ed dadmin
> config
Sections:
    1) LOCAL_DOMAINS          2) REMOTE_DOMAINS
    3) LOCAL_SERVICES        4) REMOTE_SERVICES
    5) ROUTING                6) ACCESS_CONTROL
    7) PASSWORDS             8) TDOMAIN
    9) OSITP                 10) SNA
    11) QUIT
Enter Section [1]: 2
Operations:
    1) FIRST                  2) NEXT
    3) RETRIEVE              4) ADD
    5) UPDATE                 6) DELETE
    7) NEW_SECTION           8) QUIT
Enter Operation [1]: 4
Enter editor to add/modify fields [n]? y
a
TA_RDOM                      B05
TA_DOMAINID                  BA.BANK05
TA_TYPE                      TDOMAIN
w
53
q
Perform operation [y]? <return>
Return value TAUPDATED
Buffer contents:
TA_OPERATION                  4
TA_SECTION                   2
TA_DOMAINID                  BA.BANK05
TA_RDOM                      B05
TA_TYPE                      TDOMAIN
TA_STATUS                    Update completed successfully
Operations:
    1) FIRST                  2) NEXT
    3) RETRIEVE              4) ADD
    5) UPDATE                 6) DELETE
    7) NEW_SECTION           8) QUIT
Enter Operation [4]: 7
Sections:
    1) LOCAL_DOMAINS          2) REMOTE_DOMAINS
    3) LOCAL_SERVICES        4) REMOTE_SERVICES
    5) ROUTING                6) ACCESS_CONTROL
    7) PASSWORDS             8) TDOMAIN
    9) OSITP                 10) QUIT
Enter Section [1]: 8
```

```

Operations:
    1) FIRST           2) NEXT
    3) RETRIEVE       4) ADD
    5) UPDATE         6) DELETE
    7) NEW_SECTION   8) QUIT
Enter Operation [6]: 4
Enter editor to add/modify fields [n]? y
a
TA_RDOM             B05
TA_NWADDR           0x00020401c0066d05
w
55
q
Perform operation [y]? <return>
Return value TAUPDATED
Buffer contents:
TA_OPERATION        4
TA_SECTION          8
TA_RDOM             B05
TA_NWADDR           0x00020401c0066d05
TA_STATUS           Update completed successfully
Operations:
    1) FIRST           2) NEXT
    3) RETRIEVE       4) ADD
    5) UPDATE         6) DELETE
    7) NEW_SECTION   8) QUIT
Enter Operation [4]: 8
> quit
The dmadmin program ends.

```

A.6.19 Security

If `dmadmin` is run with the application administrator's UID, it assumes a trusted user and Security is bypassed. If `dmadmin` is run with another user ID, and if the security option is enabled in the `TUXCONFIG` file, then the corresponding application password is required to start the `dmadmin` program. If standard input is a terminal, then `dmadmin` will prompt the user for the password with echo turned off. If standard input is not a terminal, the password is retrieved from the environment variable, `APP_PW`. If this environment variable is not specified and an application password is required, then `dmadmin` will fail to start.

When running with another user ID (other than the UID of the administrator) only a limited set of commands is available.

A.6.20 Environment Variables

The `dmadmin` command resets the `FIELDTBLS` and `FLDTBLDIR` environment variables to pick up the `$(TUXDIR)/udataobj/dmadmin` field table. Hence, the `TUXDIR` environment variable must be set correctly.

If the application requires security and the standard input to `dmadmin` is not from a terminal, then the `APP_PW` environment variable must be set to the corresponding application password.

The `TUXCONFIG` environment variable must be set to the pathname of the Tuxedo System/T configuration file.

A.6.21 General Diagnostics

If the `dmadmin` command is entered before the system has been booted, the following message is displayed:

```
No bulletin board exists. Only logging commands are available.
```

`dmadmin` then prompts for the corresponding commands.

If an incorrect application password is entered or is not available to a shell script through the environment, then a log message is generated, the following message is displayed, and the command terminates:

```
Invalid password entered.
```

A.6.22 Interoperability

`dmadmin` for /SNA must be installed on Tuxedo System/T R6.5. Other nodes in the same domain with an R6.5 gateway may be Tuxedo System/T R4.2.2 or later.

A.6.23 Portability

This command interpreter is supported as a Tuxedo System/T-supplied administrative tool on UNIX and Windows NT operating systems.

See Also:

`dmloadcf(1)`, `tmadmin(1)`, `dmconfig(5)`, `DMADM(5)`, `addusr(5)`, `delusr(5)` *Using the Oracle Tuxedo Domains Component*

A.7 dmconfig

Tuxedo System/T ASCII domain configuration file.

- [Description](#)
- [Definitions](#)
- [Configuration File Format](#)
- [The DM_LOCAL_DOMAINS Section](#)
- [The DM_REMOTE_DOMAINS Section](#)
- [The DM_TDOMAIN Section](#)
- [The DM_OSITP Section](#)
- [The DM_SNACRM Section](#)
- [The DM_SNASTACKS Section](#)
- [The DM_SNALINKS Section](#)

- [The DM_ACCESS_CONTROL Section](#)
- [The DM_LOCAL_SERVICES Section](#)
- [The DM_REMOTE_SERVICES Section](#)
- [The DM_ROUTING Section](#)
- [Files](#)
- [Example 1](#)
- [Example 2](#)
- [Example 3](#)

A.7.1 Description

`dmconfig` is the ASCII version of a Tuxedo System/Domain domain configuration file; it is also referred to by its environmental variable name: `DMCONFIG`. The `dmconfig` file is parsed and loaded into a binary version by the `dmloadcf` utility. The binary configuration file, called the `BDMCONFIG` file, contains information used by domain gateways to initialize the context required for communications with other domains. `dmadmin` uses the binary file (or a copy of it) in its monitoring activity. There will be one `BDMCONFIG` file for each Tuxedo System/Domain application that uses the /Domain feature.

A `DMCONFIG` file, and its binary `BDMCONFIG` counterpart, are analogous to the `UBBCONFIG` and `TUXCONFIG` files of a non-/Domain System/T application. The `DMCONFIG` file extends the definition of a non-/Domain System/T application so that the application becomes a domain.

A.7.2 Definitions

A Tuxedo System/Domain `Application` is defined as the environment described in a single `TUXCONFIG` file. A System/T Application can communicate with another System/T Application or with another TP Application via a domain gateway group. In “Tuxedo System/Domain” terms, an `Application` is the same as a TP `Domain`.

A `Gateway Group` is a collection of domain gateway processes that provide communication services with a specific type of TP `Domain`.

A `Domain Gateway` is a Tuxedo System/Domain process that relays requests and replies to another TP `Domain`.

A `Local Domain` characterizes a part of the application (set or subset of services) that is made available to other domains. A `Local Domain` is always represented by a `Domain Gateway Group`, and both terms are used as synonyms.

A `Remote Domain` is a remote application that is accessed through a `Gateway Group`. The remote application may be another Tuxedo System/Domain application or an application running under another TP system.

A `Remote Service` is a service provided by a remote domain that is made available to the local application through a `Gateway Group`.

A `Local Service` is a service of a local domain that is made available to remote domains through a `Gateway Group`.

A.7.3 Configuration File Format

The format of a domain configuration file is as follows:

- The file is made up of eight possible specification sections. Lines beginning with an asterisk (*) indicate the beginning of a specification section. Each such line contains the name of the section immediately following the *. Allowable section names are: `DM_LOCAL_DOMAINS`, `DM_REMOTE_DOMAINS`, `DM_SNACRM`, `DM_SNASTACKS`, `DM_SNALINKS`, `DM_LOCAL_SERVICES`, `DM_REMOTE_SERVICES`, `DM_ROUTING`, `DM_ACCESS_CONTROL`, `DM_OSITP`, and `DM_TDOMAIN`. The `DM_LOCAL_DOMAINS` section must precede the `DM_REMOTE_DOMAINS`.
- Parameters are generally specified by: `KEYWORD = value`. This sets `KEYWORD` to `value`. Valid keywords are described below within each section. `KEYWORD`s are reserved; they cannot be used as `value`s unless they are quoted. Lines beginning with the reserved word, `DEFAULT:`, contain parameter specifications that apply to any lines that follow them in the section in which they appear. Default specifications can be used in all sections. They can appear more than once in the same section. The format for these lines is:

```
DEFAULT: [KEYWORD1 = value1 [KEYWORD2 = value2 [...]]]
```

The values set on this line remain in effect until reset by another `DEFAULT:` line, or until the end of the section is reached. These values can also be overridden on non-`DEFAULT:` lines by placing the optional parameter setting on the line. If on a non-`DEFAULT:` line, the parameter setting is valid for that line only; lines that follow revert to the default setting. If `DEFAULT:` appears on a line by itself, all previously set defaults are cleared and their values revert to the system defaults.

If a value is *numeric*, standard C notation is used to denote the base (that is, `0x` prefix for base 16 (hexadecimal), `0` prefix for base 8 (octal), and no prefix for base 10 (decimal)). The range of values acceptable for a numeric parameter are given under the description of that parameter.

If a value is an *identifier*, standard C rules are used. An *identifier* must start with an alphabetic character or underscore and contain only alphanumeric characters or underscores. The maximum allowable length of an identifier is 30 (not including the terminating null). An identifier cannot be the same as any `KEYWORD`.

A value that is neither an integer number or an identifier must be enclosed in double quotes. Certain special characters can be escaped inside a string using a backslash. `\"` translates to a single backslash. `\\` translates to a double quote. `\n` translates to a newline. `\t` translates to a tab. `\f` translates to a form feed. `\x` (where 'x' is any character other than one of the previously mentioned special characters) translates to 'x'.

- Input fields are separated by at least one space (or tab) character.
- `#` introduces a comment. A newline ends a comment.
- Blank lines and comments are ignored.
- Comments can be freely attached to the end of any line.
- Lines are continued by placing at least one tab after the newline. Comments can not be continued.

VERSION=string_value

where *string_value* can be any value. The field is not checked by the software; it is provided simply as a place where the customer can enter a string that may have some documentation value to the application.

A.7.4 The DM_LOCAL_DOMAINS Section

This section identifies local domains and their associated gateway groups. The section must have an entry for each gateway group (Local Domain). Each entry specifies the parameters required for the domain gateway processes running in that group.

Entries have the form:

LDOM required parameters [optional parameters]

where *LDOM* is an *identifier* value used to name each local domain. *LDOM* must be unique within a particular configuration. As you will see in the description of the DM_LOCAL_SERVICES section, *LDOM* is the identifier that connects local services with a particular gateway group.

The following are the required parameters:

GWGRP = identifier

Specifies the name of the gateway server group (the name provided in the TUXCONFIG file) representing this local domain. There is a one-to-one relationship between a *DOMAINID* (see below) and the name of the gateway server group, that is, each *GWGRP* must have its own, unique *DOMAINID*.

TYPE = identifier

Is used for grouping local domain into classes. *TYPE* can be set to one of the following values: *TDOMAIN*, *OSITP* or *SNAX*. The *TDOMAIN* value indicates that this local domain can only communicate with another Tuxedo System/Domain. The *OSITP* value indicates that this local domain communicates with another TP Domain via the OSI-TP protocol. The *SNA* value indicates that this local domain communicates with an MVS/CICS region via the LU6.2 protocol. Domain types must be defined in the `$TUXDIR/udataobj/DMTYPE` file.

DOMAINID = string

Is used to identify the local domain. *DOMAINID* must be unique across both local and remote domains. The value of *string* can be a sequence of characters (for example, "BA.CENTRAL01"), or a sequence of hexadecimal digits preceded by "0x" (for example, "0x0002FF98C0000B9D6"). *DOMAINID* must be 32 octets or fewer in length. If the value is a string, it must be 32 characters or fewer (counting the trailing null).

DMTLOGDEV = string

Specifies the Tuxedo file system that contains the Domain transaction log (*DMTLOG*) for this machine. The *DMTLOG* is stored as a Tuxedo System VTOC table on the device. If this parameter is not specified (and it must not be specified if *TYPE=SNADOM*), the domain gateway group is not allowed to process requests in transaction mode. Local domains running on the same machine can share the same *DMTLOGDEV* file system, but each local domain must have its own log (a table in the *DMTLOGDEV*) named as specified by the *DMTLOGNAME* keyword (see below).

Optional parameters describe resources and limits used in the operation of domain gateways:

AUDITLOG = *string*

Specifies the name of the audit log file for this local domain. The audit log feature is activated from the `dmadmin(1)` command and records all the operations within this local domain. If the audit log feature is active and this parameter is not specified, the file `DMmmddyy.LOG` (where `mm`=month, `dd`=day, and `yy`=year) is created in the directory specified by the `$APPDIR` environment variable or the `APPDIR` keyword of the `MACHINES` section of the `TUXCONFIG` file.

BLOCKTIME = *numeric*

Specifies the maximum wait time allowed for a blocking call. The value sets a multiplier of the `SCANUNIT` parameters specified in the `TUXCONFIG` file. The value `SCANUNIT * BLOCKTIME` must be greater than or equal to `SCANUNIT` and less than 32,768 seconds. If this parameter is not specified, the default value is set to the value of the `BLOCKTIME` parameter specified in the `TUXCONFIG` file. A time-out always implies a failure of the affected request. Notice that the time-out specified for transactions in the `TUXCONFIG` will always be used when the request is issued within a transaction.

DMTLOGNAME = *identifier*

Specifies the name of the domain transaction log for this domain. This name must be unique when the same `DMTLOGDEV` is used for several local domains. If not specified, the default is the string "DMTLOG". The name must be 30 characters or less. Since transactions are not supported for SNA-type gateways, this parameter has no meaning when `TYPE=SNADOM`.

DMTLOGSIZE = *numeric*

Specifies the numeric size, in pages, of the Domain transaction log for this machine. It must be greater than 0 and less than the amount of available space on the Tuxedo file system. If not specified, the default is 100 pages. Since transactions are not supported for SNA-type gateways, this parameter has no meaning when `TYPE=SNADOM`.

MAXDATALEN = *numeric*

Specifies a maximum amount of data (in bytes) that can be sent to or from any services advertised by this local domain. There is no limit if this parameter is not specified.

MAXRDOM = *numeric*

Specifies the maximum number of connections (or dialogues if the domain is of type `OSITP`) allowed per gateway. There is no limit if this parameter is not specified.

MAXRDOMTRAN = *numeric*

Specifies the maximum number of domains that can be involved in a transaction. It must be greater than 0 and less than 32,768. If not specified, the default is 16. Since transactions are not support for SNA-type gateways, this parameter has no meaning when `TYPE=SNADOM`.

MAXTRAN = *numeric*

Specifies the maximum number of simultaneous global transactions allowed on this local domain. It must be greater than or equal to 0 and less than or equal to the `MAXGTT` parameter specified in the `TUXCONFIG` file. If not specified, the default is the value of `MAXGTT`.

MAXSENDLEN = numeric

Specifies the maximum length (in bytes) of messages sent or received by this local domain. If this parameter is set all messages sent or received will be broken up into packets of no more than MAXSENDLEN bytes. There is no limit if this parameter is not specified.

STATISTICS_INTERVAL= numeric

Specifies the time interval (in seconds) when the statistics trace is flushed to the audit log file. If this option value is greater than zero, the statistics gathering feature for the remote domain service call is enabled. If the option is not specified or set to zero, the feature is disabled. It must be greater than or equal to 0 and less than 32,768.

STATISTICSLOG_PFX= string [1...256]

Specifies the absolute pathname prefix of the statistics log file for the local domain access point. This option is associated with the statistics log feature (activated by the `dmadmin` command) which records all the service call from local domain access point to remote domain.

If this option is not specified when the statistics log feature is activated, the file `STAT_LOG_<LDOMID>.mmddyy` (where `LDOMID` is the local domain ID and `mmddyy` represents the month, day, and year respectively) is created in the path where the `$APPDIR` environment variable or the `APPDIR` parameter in the `TUXCONFIG` file `MACHINES` section specifies. The statistics log file is created when the `GWADM` server starts up.

SECURITY = value

Specifies the type of application security to be enforced. The following description applies to security in an SNA-type gateways.

The combined settings of the `SECURITY` parameters in the `UBBCONFIG` and the `DMCONFIG` files have the following effects:

- When the `DM_LOCAL_DOMAINS` Security parameter is set to `NONE` or `APP_PW`, no action is taken by the Oracle Tuxedo Mainframe Adapter for SNA gateway with regard to security.
- However, when the `UBBCONFIG` file Security parameter is set to `APP_PW`, the application password is validated by an `AUTHSVC` when clients join the application. The `AUTHSVC` is provided by the user application.

If security is to be enforced by both the local domain and the host system for each request outbound from the local domain, the following settings must be made:

- `UBBCONFIG` file Security parameter must be set to one of: `USER_AUTH`, `ACL`, or `MANDATORY_ACL`
- `DMCONFIG` file `DM_LOCAL_DOMAINS` section Security parameter must be set to `DM_USER_PW`
- `DMCONFIG` file `DM_LOCAL_DOMAINS` section Security parameter must be set to `DM_USER_PW`
- `DM_SNALINKS` Security parameter must be set to `IDENTIFY` or `VERIFY`

If security is to be enforced by both the local domain and the host system for each request inbound from the host system to the local domain, the following settings must be made:

- `UBBCONFIG` file Security parameter must be set to one of: `USER_AUTH`, `ACL`, or `MANDATORY_ACL`
- `DMCONFIG` file `DM_LOCAL_DOMAINS` section Security parameter must be set to `DM_USER_PW`
- `DM_SNALINKS` Security parameter must be set to `IDENTIFY` or `VERIFY`

For a request sent to the host system, the local principal userid is located in the domain security table and the associated remote userid, or userid and password, are put into the conversation start-up request before being sent over the LU6.2 conversation. (This occurs if `SECURITY` is set to `IDENTIFY` or `VERIFY` in the `DM_SNALINKS` section of the `DMCONFIG` file.)

For requests sent from the host system, the local domain extracts the remote userid, or userid and password, from the conversation start-up request and checks the domain security table. That table contains pairs of local principal user IDs and remote user IDs, maintained on a service-by-service basis. The remote user ID is mapped to the local principal userid. The local principal userid and password are used for further Access Control List (ACL) checking, as specified in the `UBBCONFIG` file.

When a request is received from the host system, the local domain checks the `DMCONFIG` file ACL for the local service to see if requests from the remote domain are permitted. If the `DMCONFIG` file does not contain an ACL for the local service, the service is accessible to all requests.

Therefore, if the `ATTACHSEC` level for the connection definition in the host system is `Identify` or `Verify`, the `DMCONFIG SECURITY` parameter must be set to `DM_USER_PW` so that a userid and a password are sent on the conversation start-up requests.

A.7.5 The `DM_REMOTE_DOMAINS` Section

This section identifies the known set of remote domains and their characteristics.

Entries have the form:

```
RDOM required parameters [optional parameters]
```

where *RDOM* is an *identifier* value used to identify each remote domain known to this configuration. *RDOM* must be unique within the configuration.

The following parameters are required:

TYPE = *identifier*

Is used for grouping remote domain into classes. `TYPE` can be set to one of the following values: `TDOMAIN`, `OSITP` or `SNAX`. The `TDOMAIN` value indicates that this remote domain can only communicate with another Tuxedo System/Domain. The `OSITP` value indicates that this remote domain communicates with another TP domain via the OSI-TP protocol. The `SNAX` value indicates that this remote domain communicates with an MVS/CICS region via the LU6.2 protocol.

DOMAINID = *string*

Is used to identify a remote domain. `DOMAINID` must be 32 octets or fewer in length. If the value is a string, it must be 32 characters or fewer (counting the trailing null). `DOMAINID` must be unique across remote domains. The value of *string* can be a sequence of characters or a sequence of hexadecimal digits preceded by " `0x` ".

The following parameter is optional:

CODEPAGE = "*table identifier*"

Is used to designate a bidirectional translation table for ASCII to EBCDIC conversion between a local Tuxedo application and a remote mainframe application. The table

identifier describes a file containing a translation table and must be enclosed by double quotes. The name of the file, located in the `$TUXDIR/udatajobj/codepage` directory, is a composite of the code page numbers used for the translation, for example:

```
CODEPAGE="00819x00297"
```

designates the translation table for converting ASCII CP-00819 characters to French EBCDIC CP-00297 characters, and vice versa. The translation tables can be modified. Refer to [Code Page Translation Tables](#) for complete character listings.

A.7.6 The DM_TDOMAIN Section

This section defines the addressing information required by domains of type `TDOMAIN`. This section must have an entry per local domain if requests from remote domains to local services are accepted on that local domain (gateway group), and an entry per remote domain accessible by the defined local domains.

Entries have the form:

```
DOM required parameters [optional parameters]
```

where *DOM* is an *identifier* value used to identify either a local domain (`LDOM`) or a remote domain (`RDOM`) in the `DM_LOCAL_DOMAINS` section or in the `DM_REMOTE_DOMAINS` section. The *DOM* identifier must match a previously defined *LDOM* in the `DM_LOCAL_DOMAINS` sections or *RDOM* in the `DM_REMOTE_DOMAINS` section.

The following parameter is required:

NWADDR = *string*

This parameter specifies the network address used by a local or a remote domain to accept connections from other Tuxedo System/Domain Domains. If *string* has the form `'0xhex-digits'`, it must contain an even number of valid hexadecimal digits.

The following parameter is optional:

NWIDLETIME = *numeric*

This parameter specifies the maximum time allowed for a connection to be idle (that is, unused). When this time is reached, the idle connection is terminated. The numeric value represents a time in minutes. If this keyword is not specified, then idle connections will be maintained until the gateway handling the connection is shutdown.

Notice that multiple entries for a particular domain may be defined in this table. Multiple addresses specified for a remote domain mean that the first address (the first entry in the table for the remote domain) must be used to establish the connection and the other addresses must be used as back-up addresses in case of failure of the connection setup to the first address. Multiple addresses specified for a local domain mean that multiple listening ports are available on the same or different types of networks.

A.7.7 The DM_OSITP Section

This section defines the addressing information required by domains of type `OSITP`. This section must have one entry per gateway group (local domain), and one entry per remote domain of type `OSITP`.

Entries have the form:

```
DOM required parameters [optional parameters]
```

where *DOM* is an *identifier* value used to identify a local domain (LDM) or a remote domain (RDM) in the DM_LOCAL_DOMAINS section or in the DM_REMOTE_DOMAINS section. The *DOM* identifier must match a previously defined *LDM* in the DM_LOCAL_DOMAINS sections or *RDM* in the DM_REMOTE_DOMAINS section.

The following are required parameters:

APT = string

This parameter specifies an OSI Application Process Title (APT). An APT may be a name (i.e., the Directory Name of an Application Process Title) or an object identifier (i.e., a sequence of integer values separated by periods).

AEQ = string

This parameter specifies an OSI Application Entity Qualifier (AEQ). An AEQ may be a name (i.e., the relative distinguished name of a particular Application Entity) or an integer (i.e., if the APT is an object identifier).

The following are optional parameters:

AET = string

This parameter specifies an OSI Application Entity Title (AET). An AET is formed from an Application Process Title (APT) and an Application Entity Qualifier (AEQ), i.e. in ASN.1 AET is defined as a SEQUENCE {APT, AEQ} where APT and AEQ are of type ANY. Three main formats are accepted for the value of *string* :

encoded string

This is a single value as a hexadecimal octet string which represents a valid BER encoding of the AET, e.g. AET = "0x06062B80CE0F0107".

{object identifier}, {integer}

The first element represents the APT defined as an object identifier (i.e., a sequence of integer values separated by periods) and the second element represents an AEQ defined as an integer constant, e.g., AET = "{1.3.15.0.3},{1}".

{string}, {string}

This format allows the APT and the AEQ to be defined as string constants, e.g., AET = "{BA.CENTRAL01},{Tuxedo}".

ACN = {XATMI | UDT}

This parameter specifies the object identifier of the Application Context Name (ACN) used by this domain. Current allowed application contexts are: the XATMI-ASE (XATMI) and the UDT-ASE (UDT). If this parameter is not specified, the ACN is set to the object identifier of the XATMI-ASE Application Context.

APID = integer

This parameter specifies an OSI Application Process Invocation Identifier (APID).

AEID = integer

This parameter specifies an OSI Application Entity Invocation Identifier (AEID).

PROFILE = *identifier*

This parameter specifies the OSI TP profile used by this domain and is used to determine the required OSI TP functional units. `PROFILE` can be set to one of the following values: `ATP11`, `ATP21`, `ATP31`, `ATP12`, `ATP22`, and `ATP32`. The UDT ASE application context allows the use of any of these profiles. The XATMI-ASE application context only allows profiles `ATP11`, `ATP21` and `ATP31`. Profiles `ATP11`, `ATP21` and `ATP31` use the Dialogue, Polarized Control and Handshake functional units. Profiles `ATP12`, `ATP22` and `ATP32` use the Dialogue, Shared Control, and Handshake functional units. Profiles `ATP11` and `ATP12` do not use OSI TP transactions (the Commit functional unit is not used). Profiles `ATP21` and `ATP22` require the Commit, Unchained Transactions, and Recovery functional units. Profiles `ATP31` and `ATP32` require the Commit, Chained Transactions, and Recovery functional units. By default, the `ATP21` profile is always selected.

URCH = *string*

This parameter specifies the user portion of the OSITP Recovery Context Handle. It may be required by the XAP-TP provider in order to perform recovery of distributed transactions after a communications line or system failure.

A.7.8 The DM_SNACRM Section

The `DM_SNACRM` section provides three (3) keywords used to identify the Communications Resource Manager that will provide ATMI transaction semantics between a given domain and its partners. Entries have the general form:

<CommunicationsResourceManagerName> parameters

Where <CommunicationsResourceManagerName> is the locally known name of this `SNACRM` definition to be used when referencing this `SNACRM` in subsequent sections. This name is an ASCII string 1 to 30 characters in length. The parameters are the keyword/value pairs that makeup the definition. All keywords are required for a valid `SNACRM` definition. Keywords can be in any order.

LDOM <LocalDomainName>

`LDOM` associates this `SNACRM` with a defined local domain. <LocalDomainName> is the reference to an entry in the `DM_LOCAL_DOMAINS` section. This name is an ASCII string 1 to 30 characters in length. This parameter is required. This parameter has no default.

SNACRMADDR <HexSocketAddress> or <>//host:port>

`SNACRMADDR` provides the socket address the Oracle Tuxedo Mainframe Adapter for SNA Gateway uses to communicate with the `SNACRM`. If the `SNACRM` is started independent of the Gateway, this address must be used on the `SNACRM` command line.

<HexSocketAddress> is a TCP/IP address using the `sockaddr_in` format of family,port,address: <0xFFFFPPPPAAAAAAAA>

where:

FFFF

is the hex value of the protocol family, always 0x0002 for the INET family.

PPPP

is the hex value of an unused TCP/IP port.

AAAAAAA

is the hex value of the IP address for the machine running the `SNACRM`.

Therefore if the `SNACRM` was running on a machine with an IP address of 206.189.43.13, and we wanted to use port 6000 for the `SNACRM` then `SNACMADDR` would be: `0x00021770CEBD2B0D`

This parameter is required. This parameter must contain an even number of hex characters. This parameter has no default.

A.7.9 The `DM_SNASTACKS` Section

The `DM_SNASTACKS` section provides five (5) keywords which identify the third party SNA stack that must be used for connections established between a given domain and it's partners. Entries have the general form:

<StackReference> parameters

Where *<StackReference>* is the locally known name of this stack definition to be used when referencing this stack in subsequent sections. This name is an ASCII string 1 to 30 characters in length. The parameters are the keyword/value pairs that makeup the definition. All keywords are required for a valid stack definition. Keywords can be in any order.

LOCALLU <LocalLUAlias>

`LOCALLU` provides a reference to an LU alias defined in the third party SNA stack. *<LocalLUAlias>* is the name used to identify the local LU definition as specified by the third party SNA stack configuration. This is a name that represents the end node for an LU6.2 connection. The value for this parameter is an ASCII string, 1 to 64 characters in length. This parameter is required. This parameter has no default. The third party SNA stack will require a corresponding definition for a local LU.

LTPNAME <LocalTransactionProgramName>

`LTPNAME` identifies the inbound transaction programs which will be serviced by any CRM using this stack definition. *<LocalTransactionProgramName>* is the name used to identify inbound transaction programs for which an attach will be accepted. The only useful value is an asterisk. This indicates all inbound attaches will be accepted. This parameter is required. This parameter has no default. Partial TP names are not supported. The third party SNA stack will require a corresponding definition for inbound TP names.

SNACRM <CommunicationsResourceMangerName>

`SNACRM` provides a name by which to reference the associated CRM definition. *<CommunicationsResourceMangerName>* is the name used to associate the `DM_SNACRM` definition with this `DM_SNASTACKS` entry. The value for this parameter is an ASCII string, 1 to 32 characters in length. This parameter is required. This parameter has no default.

STACKPARMS <parameters required for third party sna stack>

`STACKPARMS` provides a method for the domain gateway to pass any required parameters to the third party SNA stack. *<parameters required for third party sna stack>* is an ASCII string, 1 to 128 characters in length. Currently, the only value used is the TCP/IP hostname for the machine running the third party SNA stack. This parameter is required. This parameter has no default.

STACKTYPE={VT210 | IBM60}

This option is used to indicate which vendor SNA stack is being used. It is also used to determine the name of specific Oracle Tuxedo Mainframe Adapter for SNA system libraries. It is essential that the value of this option be coded correctly. These values are mapped to the equivalent Oracle Tuxedo Mainframe Adapter for SNA system library. Select the VT210 when the SNA stack is VTAM; select the IBM60 when the SNA stack is IBM Communications Server.

A.7.10 The DM_SNALINKS Section

This section defines the SNA Link information required by domains of type SNA. Entries have the form:

LINK parameters

Where *LINK* is an *identifier* value used to identify a connection between a local domain (LDOM) and a remote domain (RDOM). The *RDOM* identifier must match a previously defined *RDOM* in the DM_REMOTE_DOMAINS section.

The following parameters are available:

STACKREF = *string*

This required parameter defines the stack that will be used for establishment of this link. The STACKREF string is the tag that was used in a previous definition established in the DM_SNASTACKS section.

RDOM = *string*

The RDOM string must match a previous RDOM definition in the DM_REMOTE_DOMAINS section.

LSYSID = <Connection ID of remote (CICS) region>

LSYSID is the 4 character identifier that is to be used for this link. This must match the connection ID used by a partner CICS to communicate to the CRM across this link.

RSYSID = <SYSID of remote (CICS) region>

RSYSID is the 4 character remote sysid of the partner. Typically it is the sysid of a CICS region, but could also be the subsystem id of an IMS control region. This parameter must match the actual sysid of the remote partner.

RLUNAME = <Alias of APPLID for remote region>

The RLUNAME value represents an alias known to the third party SNA stack that resolves to a VTAM netname for the remote application. This would most likely be the VTAM applid for a CICS region, however it could also be an APPC/MVS LU defined for use with IMS. The value must be unique within the SNA network. *string* must be from 1 to 8 characters. This parameter is required. This parameter has no default. The third party stack configuration requires a matching definition.

MODENAME = <Mode name VTAM mode entry>

MODENAME is VTAM mode entry, defined to the third party SNA stack, to be used for this link. For a CICS link this must be compatible with the RDO session definition for the corresponding connection. For an IMS connection this must be compatible with the DLOGMOD entry on the LU definition used to access the IMS scheduler. *string* must be from 1 to 8 ASCII characters. This parameter is required. This parameter must match the third party SNA stack configuration and must be compatible with the corresponding entries defined to VTAM and/or CICS.

SECURITY = string

SECURITY_TYPE specifies the security setting in CICS/RACF or partner. Legal values are LOCAL, IDENTIFY, VERIFY, PERSISTENT or MIXIDPE. *string* must be from 1 to 10 characters. The default setting is LOCAL.

MAXSESS = number

MAXSESS is the maximum number of parallel sessions that can be started on this link. MAXSESS must be greater than or equal to four.

MINWIN = number

The minimum number of contention winners. This value is typically half the MAXSESS value.

MAXSYNCLVL = number

This value represents the maximum transaction synchronization level that can be supported over this link.

A value of zero (0) means this link is non-transactional. No synchronization will be maintained. This level can be used for sending and receiving messages from IMS via the APPC/MVS transparency interface. The default sync-level is sync-level 0.

A value of one (1) means this link will support everything supported with zero (0), in addition to:

- Outbound ATMI `tpcall()` as a CICS distributed program link request with the semantics of SYNCONRETURN.
- Inbound EXEC CICS LINK requests with the semantics of SYNCONRETURN. The program name must match the RNAME on the local service definition and the SYSID must match the LSYSID for the link.

A value of two (2) means this link will support everything supported with zero (0) and one (1) for partners able to exchange logs and compare states, in addition to:

The exchange logs and compare states function with a partner CICS.

- Outbound ATMI `tpcall()` as a CICS distributed program link request with full two phase commit transaction semantics using `tpcommit()`.
Outbound ATMI `tpconnect()` as APPC or CPIC distributed transaction processing with full two phase commit transaction semantics using `tpcommit()`
- Inbound EXEC CICS LINK requests with full two phase commit transaction semantics using Prepare Rollback and Syncpoint verbs.
Inbound APPC or CPIC conversations with full two phase commit transaction semantics using Prepare Rollback and Syncpoint verbs.

The partner must be able to negotiate a CICS style exchange logs and compare states for successful initialization of a sync-level 2 link.

Transaction support is only available at sync-level 2. Distributed Program Link can be accessed as SYNCONRETURN, that is, not transactional if the link sync-level is 1.

STARTTYPE = {auto | cold}

This option sets the recovery mode for transactional links. When set to AUTO, the system restarts using configuration and link data recovered from the in-flight transaction log. When set to COLD, the system uses configuration data taken from

the current dmconfig file and loses any in-flight link data. Changing dmconfig file parameters and performing an AUTO start results in a message warning that changed parameters are ignored until the next cold start. To force a cold start and disregard the STARTTYPE setting, delete the SNA*LOG files in \$APPDIR.

A.7.11 The DM_ACCESS_CONTROL Section

This section specifies the access control lists used by local domain. Lines in this section are of the form:

```
ACL_NAME required parameters
```

where *ACL_NAME* is a (*identifier*) name used to identify a particular access control list; it must be 15 characters or less in length.

Required parameters are:

```
ACLIST = identifier[, identifier]
```

where an **ACLIST** is composed of one or more remote domain names (**RDOM**) separated by commas. The wildcard character (*) can be used to specify that all the remote domains defined in the **DM_REMOTE_DOMAINS** section can access a local domain.

A.7.12 The DM_LOCAL_SERVICES Section

This section provides information on the services exported by each local domain. This section is optional and if it is not specified then all local domains defined in the **DM_LOCAL_DOMAINS** section accept requests to all of the services advertised by the Tuxedo System/Domain application. If this section is defined then it must be used to restrict the set of local services that can be requested from a remote domain.

Lines within this section have the form:

```
service [optional parameters]
```

where *service* is the (*identifier*) local name of the exported service, and it must be 1-15 characters in length. This name corresponds to a name advertised by one or more servers running with the local Tuxedo System/Domain application. Notice that exported services inherit the default or special properties specified for the service in an entry in the **SERVICES** section of the **TUXCONFIG** file. Some of these parameters are: **LOAD**, **PRIO**, **AUTOTRAN**, **ROUTING**, **BUFTYPE**, and **TRANTIME**.

Optional parameters are:

```
ACL = identifier
```

Specifies the name of the access control list (**ACL**) to be used by the local domain to restrict requests made to this service by remote domains. The name of the **ACL** is defined in the **DM_ACCESS_CONTROL** section. If this parameter is not specified then access control will not be performed for requests to this service.

```
API = ATMI
```

Specifies the API used by the local service. Currently the only supported value is **ATMI**. This parameter is required.

CONV = { Y | N }

Specifies whether (Y) or not (N) the local service is a conversational service. The default value is N.

LDOM = *identifier*

Specifies the name identifying the local domain exporting this service. If this keyword is not specified, then the first local domain entry in the DM_LOCAL_DOMAINS section accepts requests for this local service.

INBUFTYPE = *type[:subtype]*

Restricts the buffer type naming space of data types accepted by this service to a single buffer type. This parameter must be defined when the service is going to be used from an OSITP type gateway that uses the UDT ASE Application Context. For SNA-type gateways buffer types, see the discussion in the DM_REMOTE_SERVICES section below.

OUTBUFTYPE = *type[:subtype]*

Restricts the buffer type naming space of data types returned by this service to a single buffer type. This parameter must be defined when the service is going to be used from an OSITP type gateway that uses the UDT ASE Application Context. The FML buffer type cannot be used for OSITP type gateways. For SNA-type gateways buffer types, see the discussion in the DM_REMOTE_SERVICES section below.

RNAME = *string*

The RNAME option is the local-service name imported from a remote CICS/ESA region. This name is used by the CRM to select a local service.

When the RNAME specifies an alternate mirror transaction identifier for explicit attachment for inbound DPL requests, it must be a combination of the alternate mirror TRANSID and a CICS/ESA program name in the following format:

RNAME=AAAA:BBBBBBBB

where:

AAAA

is a 1-4 character alternate mirror TRANSID.

BBBBBBBB

is a 1-8 character CICS/ESA program name.

The colon is required to indicate the TRANSID/program name combination. The TRANSID must be composed of acceptable CICS/ESA characters:

A-Za-z0-9\$@#./-_%&Q?!"'=<>

A.7.13 The DM_REMOTE_SERVICES Section

This section provides information on services "imported" and available on remote domains. Lines within this DM_REMOTE_SERVICES section have the form:

```
service [optional parameters]
```

where *service* is the (*identifier*) name used by the local Tuxedo System/Domain application for a particular remote service. Remote services are associated with a particular remote domain.

Optional Parameters are:

AUTOTRAN = { Y | N }

Specifies whether or not a transaction must automatically be started if a request message is received that is not already in transaction mode. The default is N.

BLOCKTIME = numeric

Specifies the maximum wait time allowed for a reply to this remote service. The value sets a multiplier of the SCANUNIT parameters specified in the TUXCONFIG file. The value SCANUNIT * BLOCKTIME must be greater than or equal to SCANUNIT and less than 32,768 seconds. A time-out always implies a failure of the affected transaction or request.

CONV = { Y | N }

Specifies whether (Y) or not (N) the remote service is a conversational service. The default value is N.

FUNCTION = {APPC|DPL}

Enables outbound Tuxedo service requests to map to APPC transaction programs or CICS programs. The default value APPC indicates the remote service is a transaction program that may or may not be running under CICS. The DPL value indicates the remote service maps to a program running under CICS.

LDOM = identifier

Specifies the name of a local domain in charge of routing requests to this remote service. The gateway group associated with the local domain advertises *service* in the Tuxedo System/Domain Bulletin Board. If this parameter is not specified then all the local domains will be able to accept requests to this remote service. The service request will be then redirected to a remote domain of the same type (see RDOM keyword below).

LOAD = integer

Specifies that the remote service imposes a load of integer units. The value of LOAD can be between 1 and 32767 inclusive. If not specified, the default is 50. A higher number indicates a greater load.

INBUFTYPE = type[:subtype]

Restricts the buffer type naming space of data types accepted by this service to a single buffer type. This parameter must be defined when the service is going to be used from an OSITP type gateway that uses the UDT ASE Application Context. The FML buffer type cannot be used for OSITP type gateways.

OUTBUFTYPE = type[:subtype]

Restricts the buffer type naming space of data types returned by this service to a single buffer type. This parameter must be defined when the service is going to be used from an OSITP type gateway that uses the UDT ASE Application Context. The FML buffer type cannot be used for OSITP type gateways.

PRIO = integer

Specifies the dequeuing priority of service requests to this remote service. The value of PRIO must be greater than 0 and less than or equal to 100, with 100 being the highest priority. The default is 50.

RDOM = *identifier*

Specifies the name of the remote domain responsible for the actual execution of this service. If this parameter is not specified and a routing criteria (see below `ROUTING` keyword) is not specified, then the local domain assumes that any remote domain of the same type accepts this service and it selects a known domain (a domain to which a connection already exists) or remote domain from the `\DM_REMOTE_DOMAINS` section.

RNAME = *string*

Specifies the actual service name expected by the remote domain. If this parameter is not specified, the remote service name is the same as the name specified in `service`.

The `RNAME` option is the name of the host `TP_NAME`. For non-CICS systems, this name can be up to 64 characters in length. For CICS systems, this name is the trans-id name for APPC-defined requests and the program name for DPL requests. CICS trans-id names cannot exceed four characters and CICS program names cannot exceed eight characters. The `RNAME` option must observe these requirements.

When the `RNAME` specifies an alternate mirror transaction identifier for explicit attachment to outbound DPL requests, it must be a combination of the alternate mirror `TRANSID` and an advertised remote CICS/ESA program name in the following format:

RNAME=AAA:BBBBBBB

where:

AAA

is a 1-4 character alternate mirror `TRANSID`.

BBBBBBB

Is a 1-8 character CICS/ESA program name.

The colon is required to indicate the `TRANSID`/program name combination. The `TRANSID` must be composed of acceptable characters recognized in CICS/ESA identifiers: A-Za-z0-9\$@#./-_%&Q?!"'=";<>

ROUTING = *identifier*

When more than one remote domain offers the same service, a local domain can perform data dependent routing if this optional parameter is specified. The `identifier` specifies the name of the routing criteria used for this data dependent routing. If not specified, data dependent routing is not done for this service. `identifier` must be 15 characters or less in length. If multiple entries exist for the same service name but with different `RDOM` parameters, the `ROUTING` parameter must be the same for all of these entries.

TRANTIME = *integer*

specifies the default time-out value in seconds for a transaction automatically started for the associated service. The value must be greater than or equal to 0 and less than 2147483648. The default is 30 seconds. A value of 0 implies the maximum time-out value for the machine.

A.7.14 The DM_ROUTING Section

This section provides information for data dependent routing of /T Domain service requests using FML, VIEW, X_C_TYPE, and X_COMMON typed buffers. Lines within the DM_ROUTING section have the form:

```
CRITERION_NAME required parameters
```

where *CRITERION_NAME* is the (*identifier*) name of the routing entry that was specified on the services entry. *CRITERION_NAME* must be 15 characters or less in length.

Required parameters are:

FIELD = *identifier*

Specifies the name of the routing field. It must be 30 characters or less. This field is assumed to be a field name that is identified in an FML field table (for FML buffers) or an FML view table (for VIEW, X_C_TYPE, or X_COMMON buffers). The FLDTBLDIR and FIELDTBLS environment variables are used to locate FML field tables, and the VIEWDIR and VIEWFILES environment variables are used to locate FML view tables.

RANGES = *string*

Specifies the ranges and associated remote domain names (RDOM) for the routing field. *string* must be enclosed in double quotes. The format of *string* is a comma-separated ordered list of range/RDOM pairs (see EXAMPLES below).

A range is either a single value (signed numeric value or a character string in single quotes) or a range of the form “lower - upper” (where lower and upper are both signed numeric values or character strings in single quotes). Note that “lower” must be less than or equal to “upper”. To embed a single quote in a character string value (as in O’Brien, for example), the single quote must be preceded by two backslashes (‘O\\’Brien’). The value MIN can be used to indicate the minimum value for the data type of the associated FIELD; for strings and arrays, it is the null string; for character fields, it is 0; for numeric values, it is the minimum numeric value that can be stored in the field. The value MAX can be used to indicate the maximum value for the data type of the associated FIELD; for strings and arrays, it is effectively an unlimited string of octal-255 characters; for a character field, it is a single octal-255 character; for numeric values, it is the maximum numeric value that can be stored in the field. Thus, “MIN - -5” is all numbers less than or equal to -5 and “6 - MAX” is all numbers greater than or equal to 6. The meta-character “*” (wild-card) in the position of a range indicates any values not covered by the other ranges previously seen in the entry; only one wild-card range is allowed per entry and it must be last (ranges following it will be ignored).

The routing field can be of any data type supported in FML. A numeric routing field must have numeric range values and a string routing field must have string range values.

The routing field can be of any data type supported in FML. A numeric routing field must have numeric range values and a string routing field must have string range values.

String range values for string, array, and character field types must be placed inside a pair of single quotes and can not be preceded by a sign. Short and long integer values are a string of digits, optionally preceded by a plus or minus sign. Floating point numbers are of the form accepted by the C compiler or atof(): an optional sign, then a string of digits optionally

containing a decimal point, then an optional e or E followed by an optional sign or space, followed by an integer.

When a field value matches a range, the associated `RDOM` value specifies the remote domain to which the request must be routed. A `RDOM` value of "*" indicates that the request can go to any remote domain known by the gateway group.

Within a range/`RDOM` pair, the range is separated from the `RDOM` by a ":".

```
BUFTYPE = ~type1[:subtype1[, subtype2 . . . ]]  
[; type2[:subtype3[, . . . ]]] . . . ~
```

Is a list of types and subtypes of data buffers for which this routing entry is valid. The types are restricted to be either `FML`, `VIEW`, `X_C_TYPE`, or `X_COMMON`. No subtype can be specified for type `FML` and subtypes are required for the other types ("*" is not allowed). Duplicate type/subtype pairs can not be specified for the same routing criterion name; more than one routing entry can have the same criterion name as long as the type/subtype pairs are unique. This parameter is required. If multiple buffer types are specified for a single routing entry, the data types of the routing field for each buffer type must be the same.

If the field value is not set (for `FML` buffers), or does not match any specific range and a wild-card range has not been specified, an error is returned to the application process that requested the execution of the remote service.

A.7.15 Files

The `BDMCONFIG` environment variable is used to find the `BDMCONFIG` configuration file.

A.7.16 Example 1

The following configuration file defines a 5-site domain configuration. The example shows 4 Bank Branch domains communicating with a Central Bank Branch. Three of the Bank Branches run within other Tuxedo System/Domain domains. The fourth Branch runs under the control of another TP Domain and OSI-TP is used in the communication with that domain.

```
# Tuxedo DOMAIN CONFIGURATION FILE FOR THE CENTRAL BANK  
#  
#  
*DM_LOCAL_DOMAINS  
# <local domain name> <Gateway Group name> <domain type> <domain id>  
<log device>  
#                               [<audit log>] [<blocktime>]  
#                               [<log name>] [<log offset>] [<log size>]  
#                               [<maxrdom>] [<maxrdtran>] [<maxtran>]  
#                               [<maxdatalen>] [<security>]  
#                               [<tuxconfig>] [<tuxoffset>]  
#  
#  
DEFAULT: SECURITY = NONE  
c01      GWGRP = bankg1  
         TYPE = TDOMAIN  
         DOMAINID = "BA.CENTRAL01"
```

```

        DMTLOGDEV = "/usr/apps/bank/DMTLOG"
        DMTLOGNAME = "DMTLG_C01"
c02    GWGRP = bankg2
        TYPE = OSITP
        DOMAINID = "BA.CENTRAL01"
        DMTLOGDEV = "/usr/apps/bank/DMTLOG"
        DMTLOGNAME = "DMTLG_C02"
        URCH = "ABCD"

#
*DM_REMOTE_DOMAINS
#<remote domain name> <domain type> <domain id>
#
b01    TYPE = TDOMAIN
        DOMAINID = "BA.BANK01"
b02    TYPE = TDOMAIN
        DOMAINID = "BA.BANK02"
b03    TYPE = TDOMAIN
        DOMAINID = "BA.BANK03"
b04    TYPE = OSITP
        DOMAINID = "BA.BANK04"
        URCH = "ABCD"

*DM_TDOMAIN
#
# <local or remote domain name>    <network address>
#
# Local network addresses
c01    NWADDR = "0x0002ff98c00b9d6d"
c01    NWADDR = "newyork01.65432"
# Remote network addresses
b01    NWADDR = "0x00020401c00b6d05"
b02    NWADDR = "dallas.65432"
b03    NWADDR = "0x00021094c00b6d9c"

*DM_OSITP
#
#<local or remote domain name> <apt> <aeq>
#                                     [<aet>] [<acn>] [<apid>] [<aeid>]
#                                     [<profile>]
#
c02    APT = "BA.CENTRAL01"
        AEQ = "Tuxedo.R.4.2.1"
        AET = "{1.3.15.0.3},{1}"
        ACN = "XATMI"
b04    APT = "BA.BANK04"
        AEQ = "Tuxedo.R.4.2.1"
        AET = "{1.3.15.0.4},{1}"
        ACN = "XATMI"

*DM_LOCAL_SERVICES
#<service_name> [<Local Domain name>] [<access control>] [<exported
svcname>]
#                                     [<inbuftype>] [<outbuftype>]
#
open_act    ACL = branch
close_act   ACL = branch
credit
debit

```

```

balance
loan          LDOM = c02          ACL = loans
*DM_REMOTE_SERVICES
#<service_name>    [<Remote domain name>] [<local domain name>]
#                  [<remote svcname>] [<routing>] [<conv>]
[<trantime>]
#                  [<inbuftype>] [<outbuftype>]
#
tlr_add LDOM = c01  ROUTING = ACCOUNT
tlr_bal LDOM = c01  ROUTING = ACCOUNT
tlr_add RDOM = b04  LDOM = c02  RNAME = "TPSU002"
tlr_bal RDOM = b04  LDOM = c02  RNAME = "TPSU003"
*DM_ROUTING
# <routing criteria> <field> <typed buffer> <ranges>
#
ACCOUNT FIELD = branchid BUFTYPE = "VIEW:account"
RANGES = "MIN - 1000:b01, 1001-3000:b02, *:b03"
*DM_ACCESS_CONTROL
#<acl name> <Remote domain list>
#
branch ACLIST = b01, b02, b03
loans ACLIST = b04

```

A.7.17 Example 2

This example shows the Tuxedo System/Domain Configuration file required at one of the Bank Branches (BANK01).

```

#
#Tuxedo DOMAIN CONFIGURATION FILE FOR A BANK BRANCH
#
#
*DM_LOCAL_DOMAINS
#
b01      GWGRP = auth
         TYPE = TDOMAIN
         DOMAINID = "BA.BANK01"
         DMTLOGDEV = "/usr/apps/bank/DMTLOG"
*DM_REMOTE_DOMAINS
#
c01      TYPE = TDOMAIN
         DOMAINID = "BA.CENTRAL01"
*DM_TDOMAIN
#
b01      NWADDR = "0x00021094c00b689c"
c01      NWADDR = "0x0002ff98c00b9d6d"
*DM_LOCAL_SERVICES
#
tlr_add  ACL = central
tlr_bal  ACL = central
*DM_REMOTE_SERVICES
#
OPA001   RNAME = "open_act"
CLA001   RNAME = "close_act"

```

```

CRD001      RNAME = "credit"
DBT001      RNAME = "debit"
BAL001      RNAME = "balance"
DM_ACCESS_CONTROL
#
central     ACLIST = c01

```

A.7.18 Example 3

This example shows the configuration file entries for an Oracle Tuxedo Mainframe Adapter for SNA application:

```

#=====
# DMCONFIG
#   Application Domain Gateway Test Configuration
#
# See also
#   See $(TOP)/Makefile for more information.
#
# @(#)SNA Devel apps/simpsna DMCONFIG 1.6 98/03/03 15:35:29
# Copyright 1997, BEA Systems, Inc., all rights reserved.
#-----
*DM_LOCAL_DOMAINS
simpsnad
    GWGRP=GROUP2
    TYPE=SNAX
    DOMAINID="simpsnad"
    BLOB_SHM_SIZE=1000000
    DMTLOGDEV=<your Tuxedo filesystem device and name for
    DMTLOG>
#example DMTLOGDEV="/home/me/bin/DMTLOG"

*DM_REMOTE_DOMAINS

SIMPSNAG TYPE=SNAX DOMAINID="SIMPSNAG"
*DM_SNACRM

    simpcrm          SNACRMADDR="<your Host Socket Listen Address>"
                    LDOM="simpsnad"
#example SNACRMADDR="0x00021770cfbd2b0d" INET family 0x0002 port 6000 host
207.189.43.13 or SNACRMADDR="//207.189.43.13:6000

*DM_SNASTACKS

simpstk
    SNACRM="simpcrm"
    STACKTYPE=<SNACRM Stack Library Named Token>
    LOCALLU=<Local LU definition specified in
    stack product>
    LTPNAME="*"
    STACKPARMS=<Parameters passed to Stack
    Product>

#example STACKTYPE="VT210"

```

```
#          LOCALLU="BEAAPPL1"
#          STACKPARMS="testhp" Name of the host machine

*DM_SNALINKS

simplk1   STACKREF="simpstk"
          RDOM="SIMPSNAG"
          LSYSID=<Connection ID of remote (CICS)
          region>
          RSYSID=<SYSID of remote (CICS) region>
          RLUNAME=<Alias of Applid for remote region>
          MODENAME=<Mode name VTAM mode entry>
          SECURITY="LOCAL"
          STARTTYPE="COLD"
          MAXSESS=<Total Session number>
          MINWIN=<Session Local Winners>
          MAXSYNCLVL=<0|1|2 Maximum Syncpoint Level>

#example LSYSID="BEA"
#          RSYSID="TEST"
#          RLUNAME="CICSTEST"
#          MODENAME="SMSNA100"
#          MAXSESS=10
#          MINWIN=5
#          MAXSYNCLVL=2

*DM_LOCAL_SERVICES

MIRROR LDOM="simpsnad"
        CONV=N
        RNAME="MIRRORSERV"
        INBUFTYPE="STRING"
        OUTBUFTYPE="STRING"
        API="ATMI"

*DM_REMOTE_SERVICES

SIMPDPD  AUTOTRAN=N
          LDOM="simpsnad"
          RDOM=SIMPSNAG
          CONV=N
          RNAME="TOUPDPLS"
          INBUFTYPE="STRING"
          OUTBUFTYPE="STRING"
          API="ATMI"
          FUNCTION="DPL"

SIMPDTP  AUTOTRAN=N
          LDOM="simpsnad"
          RDOM=SIMPSNAG
          CONV=N
          RNAME="DTPS"
          INBUFTYPE="STRING"
          OUTBUFTYPE="STRING"
```

```
API="ATMI"
FUNCTION="APPC"
```

 **Note:**

- `build_dgw(1), dmadmin(1), tmboot(1), tmshutdown(1), dmloadcf(1), dmunloadcf(1)`
- `dmgwopts(5), GWADM(5), DMADM(5)`
- *Using the Oracle Tuxedo Domains Component*

A.8 dmloadcf

Parse a `DMCONFIG` file and load binary `BDMCONFIG` configuration file.

- [Synopsis](#)
- [Description](#)
- [Portability](#)
- [Environment Variables](#)
- [Examples](#)
- [Diagnostics](#)

A.8.1 Synopsis

```
dmloadcf [-c] [-n] [-y] [-b blocks]{dmconfig_file | - }
```

A.8.2 Description

`dmloadcf` reads a file or the standard input that is in `DMCONFIG` syntax, checks the syntax, and optionally loads a binary `BDMCONFIG` configuration file. The `BDMCONFIG` environment variable points to the path name of the `BDMCONFIG` file where the information must be stored.

`dmloadcf` prints an error message if it finds any required section of the `DMCONFIG` file missing. If a syntax error is found while parsing the input file, `dmloadcf` exits without performing any updates to the `BDMCONFIG` file.

`dmloadcf` requires the existence of the `$TUXDIR/udataobj/DMTYPE` file. This file defines the valid domain types. If this file does not exist, `dmloadcf` exits without performing any updates to the `BDMCONFIG` file.

The effective user identifier of the person running `dmloadcf` must match the `UID` in the `RESOURCES` section of the `TUXCONFIG` file.

The `-c` option to `dmloadcf` causes the program to print minimum IPC resources needed for each local domain (gateway group) in this configuration. The `BDMCONFIG` file is not updated.

The `-n` option to `dmloadcf` causes the program to do only syntax checking of the ASCII `DMCONFIG` file without actually updating the `BDMCONFIG` file.

After syntax checking, `dmloadcf` checks to see if the file pointed to by `BDMCONFIG` exists, is a valid Tuxedo System file system, and contains `BDMCONFIG` tables. If these conditions are not true, the user is prompted to create and initialize the file with

```
Initialize BDMCONFIG file: path [y, q]?
```

where `path` is the complete file name of the `BDMCONFIG` file. Prompting is suppressed if the standard input or output are not terminals, or if the `-y` option is specified on the command line. Any response other than “y” or “Y” will cause `dmloadcf` to exit without creating the configuration file.

If the `BDMCONFIG` file is not properly initialized, and the user has given the go-ahead, `dmloadcf` creates the Tuxedo file system and then creates the `BDMCONFIG` tables. If the `-b` option is specified on the command line, its argument is used as the number of blocks for the device when creating the Tuxedo file system. If the value of the `-b` option is large enough to hold the new `BDMCONFIG` tables, `dmloadcf` will use the specified value to create the new file system; otherwise, `dmloadcf` will print an error message and exit. If the `-b` option is not specified, `dmloadcf` will create a new file system large enough to hold the `BDMCONFIG` tables. The `-b` option is ignored if the file system already exists. The `-b` option is highly recommended if `BDMCONFIG` is a raw device (that has not been initialized) and must be set to the number of blocks on the raw device. The `-b` option is not recommended if `BDMCONFIG` is a regular UNIX file.

If the `BDMCONFIG` file is determined to already have been initialized, `dmloadcf` ensures that the local domain described by that `BDMCONFIG` file is not running. If a local domain is running, `dmloadcf` prints an error message and exits. Otherwise, `dmloadcf`, to confirm that the file must be overwritten, prompts the user with:

```
“Really overwrite BDMCONFIG file [y, q]?”
```

Prompting is suppressed if the standard input or output are not a terminal or if the `-y` option is specified on the command line. Any response other than “y” or “Y” will cause `dmloadcf` to exit without overwriting the file.

If the `SECURITY` parameter is specified in the `RESOURCES` section of the `TUXCONFIG` file, then `dmloadcf` will flush the standard input, turn off terminal echo and prompt the user for an application password as follows:

```
Enter Application Password?
```

The password is truncated to 8 characters. The option to load the ASCII `DMCONFIG` file via the standard input (rather than a file) cannot be used when this `SECURITY` parameter is turned on. If the standard input is not a terminal, that is, if the user cannot be prompted for a password (as with a `here` file, for example), then the environment variable `APP_PW` is accessed to set the application password. If the environment variable `APP_PW` is not set with the standard input not a terminal, then `dmloadcf` will print an error message, generate a log message and fail to load the `BDMCONFIG` file.

Assuming no errors, and if all checks have passed, `dmloadcf` loads the `DMCONFIG` file into the `BDMCONFIG` file. It will overwrite all existing information found in the `BDMCONFIG` tables.

A.8.3 Portability

This command is supported as a Tuxedo-supplied administrative tool on UNIX and Windows NT operating systems.

A.8.4 Environment Variables

The environment variable `APP_PW` must be set for applications that require security (the `SECURITY` parameter in the `TUXCONFIG` file is set to `APP_PW`) and `dmloadcf` is run with something other than a terminal as the standard input.

The `BDMCONFIG` environment variable must point to the `BDMCONFIG` file.

A.8.5 Examples

The following example shows how a binary configuration file is loaded from the `bank.dmconfig` ASCII file. The `BDMCONFIG` device is created (or re-initialized) with 2000 blocks:

```
dmloadcf -b 2000 -y bank.dmconfig
```

A.8.6 Diagnostics

If an error is detected in the input, the offending line is printed to standard error along with a message indicating the problem. If a syntax error is found in the `DMCONFIG` file or the system is currently running, no information is updated in the `BDMCONFIG` file and `dmloadcf` exits with exit code 1.

If `dmloadcf` is run on an active node, the following error message is displayed:

```
*** dmloadcf cannot run on an active node ***
```

If `dmloadcf` is run by a person whose effective user identifier doesn't match the `UID` specified in the `TUXCONFIG` file, the following error message is displayed:

```
*** UID is not effective user ID ***
```

Upon successful completion, `dmloadcf` exits with exit code 0. If the `BDMCONFIG` file is updated, a `userlog` message is generated to record this event.

See Also:

- `dmunloadcf(1)`, `dmconfig(5)`, `ubbconfig(5)`
- *Using the Oracle Tuxedo Domains Component*

A.9 dmunloadcf

Unload binary `BDMCONFIG` domain configuration file

- [Synopsis](#)
- [Description](#)
- [Portability](#)
- [Examples](#)
- [Diagnostics](#)

A.9.1 Synopsis

```
dmunloadcf
```

A.9.2 Description

`dmunloadcf` translates the `BDMCONFIG` configuration file from the binary representation into ASCII. This translation is useful for transporting the file in a compact way between machines with different byte ordering and backing up a copy of the file in a compact form for reliability. The ASCII format is the same as is described in `dmconfig(5)`.

`dmunloadcf` reads values from the `BDMCONFIG` file pointed to by the `BDMCONFIG` environment variable and writes them to its standard output.

A.9.3 Portability

This command is supported as a Tuxedo-supplied administrative tool on UNIX and Windows NT operating systems.

A.9.4 Examples

To unload the configuration in `/usr/tuxedo/BDMCONFIG` into the file `bdmconfig.backup`:

```
BDMCONFIG=/usr/tuxedo/BDMCONFIG dmunloadcf > bdmconfig.backup
```

A.9.5 Diagnostics

`dmunloadcf` checks that the file pointed to by the `BDMCONFIG` environment variable exists, is a valid Tuxedo file system, and contains `BDMCONFIG` tables. If any of these conditions is not met, `dmunloadcf` prints an error message and exits with error code 1. Upon successful completion, `dmunloadcf` exits with exit code 0.

**See Also:**

- [dmloadcf\(1\)](#), [dmconfig\(5\)](#)
- [Using the Oracle Tuxedo Domains Component](#)

A.10 GWADM

/Domain gateway administrative server.

- [Synopsis](#)
- [Description](#)
- [Portability](#)
- [Interoperability](#)
- [Examples](#)

A.10.1 Synopsis

```
GWADM SRVGRP = "identifier" SRVID = "number" REPLYQ = "N"
      CLOPT = "-A -- [-a { on | off } ] [-s services ]
      [-t { on | off } ]"
```

A.10.2 Description

The gateway administrative server (GWADM) is a Tuxedo-supplied server that provides administrative functions for a /Domain gateway group.

GWADM must be defined in the SERVERS section of the UBBCONFIG file as a server running within a particular gateway group, that is, SRVGRP must be set to the corresponding GRPNAME tag specified in the GROUPS section. The SVRID parameter is also required and its value must consider the maximum number of gateways allowed within the gateway group.

There must be only one instance of a GWADM per /Domain gateway group, and it must NOT be part of the MSSQ defined for the gateways associated with the group. Also, GWADM must have the REPLYQ attribute set to N.

The CLOPT option is a string of command line options that is passed to the GWADM when it is booted. This string has the following format:

```
CLOPT="-A -- <gateway group runtime
      parameters>"
```

The following runtime parameters are recognized for a gateway group:

-a { on | off }

This option turns *off* or *on* the audit log feature for this local domain. The default is *off*. The `dmadmin` program can be used to change this setting while the gateway group is running (see [dmadmin\(1\)](#)).

-s *services*

Specifies the remote *services* that must be initially offered by the domain gateway. The specifications for these services are found in the DMCONFIG file. For example, the specification

```
-s x,y,z
```

implies that the gateway must initially advertise remote services *x*, *y*, and *z*. Spaces are not allowed between commas and the `-s` option may appear several times.

-t { *on* | *off* }

This option turns *off* or *on* the statistics gathering feature for the local domain. The default is *off*. The `dmadmin` program can be used to change this setting while the gateway group is running (see `dmadmin(1)`).

The GWADM server must be booted before the corresponding gateways.

A.10.3 Portability

This server is supported on Tuxedo-supplied servers, using UNIX System and Windows NT operating systems.

A.10.4 Interoperability

The initial release of SNA-type gateways can only be installed on a node running Tuxedo.

A.10.5 Examples

The following example illustrates the definition of the administrative server in the UBBCONFIG file.

```
#
*GROUPS
DMADMGRP   GRPNO=1
gwgrp      GRPNO=2
#
*SERVERS
DMADM SRVGRP="DMADMGRP" SRVID=1001 REPLYQ=N RESTART=Y GRACE=0
GWADM SRVGRP="gwgrp" SRVID=1002 REPLYQ=N RESTART=Y GRACE=0
        CLOPT="-A -- -a on -t on"
SNACRM SRVGRP="gwgrp" SRVID=1003 CLOPT="-A--//host:6000 gwgrp"
GWSNAX SRVGRP="gwgrp" SRVID=1004 RQADDR="gwgrp" REPLYQ=N
```

 **See Also:**

- `dmadmin(1)`, `tmboot(1)`
- `dmconfig(5)`, `DMADM(5)`, `servopts(5)`, `ubbconfig(5)`
- *Using the Oracle Tuxedo Domains Component*

A.11 GWSNAX

This is the gateway server process for Oracle Tuxedo Mainframe Adapter for SNA.

- [Synopsis](#)
- [Description](#)
- [Portability](#)
- [Interoperability](#)
- [Examples](#)

A.11.1 Synopsis

```
GWSNAX SRVGRP = "identifier" SRVID = "number" REPLYQ = "N"
                CLOPT = "-A -- [-m -n {type:min:max} -N -t {number} -c {number}
-T {number} -u {keyfile} -b {number} -M]"
...
```

A.11.2 Description

The GWSNAX server provides Tuxedo functions for an Oracle Tuxedo Mainframe Adapter for SNA Gateway group.

GWSNAX must be defined in the `SERVERS` section of the `UBBCONFIG` file as a server running within a particular gateway group; that is, `SRVGRP` must be set to the corresponding `GRPNAME` tag specified in the `GROUPS` section. The `SVRID` parameter is also required and its value must consider the maximum number of gateways allowed within the gateway group. The GWSNAX definition must not precede its associated CRM server definition in the `UBBCONFIG` file.

There must be only one instance of a GWSNAX per Oracle Tuxedo Mainframe Adapter for SNA Gateway group, and it must NOT be part of the MSSQ defined for the gateways associated with the group.

The `CLOPT` option is a string of command line options that is passed to the GWSNAX when it is booted. This string has the following format:

```
CLOPT="-A -- <gateway runtime parameters>"
```

The following runtime parameters are recognized for a gateway:

-m

This option specifies that userid mapping is bypassed. Any userid mapping already defined in the `DMCONFIG` file is preserved, but is not in effect.

-n {type:min:max}

Establishes that encryption is in effect for this client process. `type` is the encryption type. Currently, the valid entries are GPE and TLS. The `min` and `max` values designate the minimum and maximum number of bits to be used for encryption. This level is 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 GPE:

Any number is acceptable, but the negotiated values resolve to 0, 56, or 128. This level is used during the negotiation between the CRM and GWSNAX. The level specified must be supported by the security add-on package used.

For TLS:

The `max` and `min` define the acceptable range of TLS cipher suite, including 40, 56, 112, 128 and 256 bits kinds of cipher suite. `min` must be less than or equal to 256.

-N

This option indicates GWSNAX does not align data during data conversion.

This option is only used for COBOL program in ARTDPL. Usually this kind of program is migrated from mainframe CICS via workbench. Unlike standard ATMI COBOL application, this program does not use View32 generated copybook (which has alignment), instead, it uses copybooks (which has no alignment) migrated from mainframe CICS.

However, we do not recommend you use this method to resolve the non-alignment View32 issue for ART CICS integration because it results in additional overhead time.

Instead, we recommend you use the method of specifying viewc32 compile flag "`-fpack-struct=1`" to resolve this issue.

-t {number}

This option indicates the type of character string transformation the gateway performs. (Refer to the following table A-10 for values.)

For Tuxedo ATMI application, set the parameter "`-t1`" in `GWSNAX CLOPT` list, for every string type field of a View32 needs an addition null terminator.

For ART CICS integration, this parameter can be omitted or set as "`-t 4`", for the null terminator is not needed.

-c {number}

This option indicates the decimal encoding value of the character to be converted to/from the trailing null character in the string transformation the gateway performs. The value of this option must be an integer in the decimal range of 0 to 255. This option only takes effect when the `-t` option is specified with value 1 or 2.

-T {number}

This option indicates the inbound transaction timeout value.

-u {keyfile}

Establishes that process authentication is in effect for communications between this process and the CRM. The `keyfile` is the location 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 must be protected.

-b {number}

Heart beat switch option. Sometimes a router may break TCP connections between SNAX and CRM (for example, long idle time with no transaction. The router breaks connections as a given policy), the `-b` option is used to avoid this. This option is followed by a numeric value as the polling interval. The minimum/default value is 30 seconds.

-M

Enables Oracle Tuxedo Metadata Repository support.

**Note:**

The Oracle Tuxedo Metadata repository server, [TMMETADATA \(5\)](#), must be configured, and must be started before the `GWSNAX` server.

Table A-10 A-10 C to COBOL String Transformation

CLOPT -t Parameter Value	Tuxedo Application Language	Host Application Language
Not Set	No string transformation established	
1	C	COBOL
2	COBOL	C
3	C	C
4	COBOL	COBOL

A.11.3 Portability

Refer to the Oracle Tuxedo Mainframe Adapter for SNA *Release Notes* for a complete listing of compatible operating systems.

A.11.4 Interoperability

Refer to the Oracle Tuxedo Mainframe Adapter for SNA *Release Notes* for a complete listing of supported platforms.

A.11.5 Examples

The following example illustrates the definition of the administrative server in the `UBBCONFIG` file.

```
#
*GROUPS
DMADMGRP GRPNO=1
gwgrp GRPNO=2
```

```
#
*SERVERS
DMADM SRVGRP="DMADMGRP" SRVID=1001 REPLYQ=N RESTART=Y GRACE=0
GWADM SRVGRP="gwgrp" SRVID=1002 REPLYQ=N RESTART=Y GRACE=0
      CLOPT="-A -- -a on -t on"
SNACRM SRVGRP="gwgrp" SRVID=1003 CLOPT="-A--//host:6000 gwgrp"
GWSNAX SRVGRP="gwgrp" SRVID=1004 RQADDR="gwgrp" REPLYQ=N
      CLOPT="-- -t 1"
```

See Also:

- [dmadmin\(1\)](#), [tmboot\(1\)](#)
- [dmconfig\(5\)](#), [DMADM\(5\)](#), [servopts\(5\)](#), [ubbconfig\(5\)](#)
- *Using the Oracle Tuxedo Domains Component*

A.12 modusr

Modify a remote user password.

- [Synopsis](#)
- [Description](#)
- [Portability](#)
- [Diagnostics](#)
- [Examples](#)

A.12.1 Synopsis

```
modusr -d <local domain> ID -R <remote domain ID> -u <remote username>
```

A.12.2 Description

`modusr` can only be executed as a subcommand of `dmadmin(1)`. The purpose of this page is to describe options for the subcommand and to show an example.

The subcommand allows the administrator to modify passwords in the remote password table. The administrator is prompted for the remote password.

The table entries modified are used for passing remote user names and passwords to remote SNA domains when the application is using SNA-type gateways and `SECURITY` is set to `USER_AUTH`, `ACL`, or `MANDATORY_ACL` in the `ubbconfig` file and `SECURITY` is set to `DM_USER_PW` in the `DMCONFIG` file.

The following options are available:

-d <local domain ID>

This is the name of the local domain gateway with which the ids and passwords are associated. This is the same ID as the one used when creating the domain definitions either in the `DMCONFIG` file or through the Graphical Administrative Interface.

-R <remote domain ID>

This is the name of the remote domain gateway with which the ids and passwords are associated. This is the same ID as the one used when creating the domain definitions either in the `DMCONFIG` file or through the Graphical Administrative Interface.

-u <remote username>

The remote user whose password is being modified.

Before running this subcommand the application must be configured using either the Graphical Administrative Interface or `tmloadcf(1)` and `dmloadcf(1)`. `dmadmin modusr` may be run on any active node.

A.12.3 Portability

This subcommand is available on the latest version of Tuxedo, as documented for this release of Oracle Tuxedo Mainframe Adapter for SNA.

A.12.4 Diagnostics

The `dmadmin modusr` subcommand exits with a return code of 0 upon successful completion.

A.12.5 Examples

```
modusr -d tux -R cics -u CICSUSR /*modifies remote user's password
                               sent to CICS. The administrator
                               is prompted for the password*/
```



See Also:

- [delusr\(5\)](#), [addusr\(5\)](#)

A.13 tmadmin

- [tmadmin\(1\)](#)
- [tmadmin Command Related to TMA](#)

A.13.1 tmadmin(1)

- [Name](#)
- [Synopsis](#)
- [Description](#)

A.13.1.1 Name

tmadmin—Oracle Tuxedo bulletin board command interpreter.

A.13.1.2 Synopsis

```
tmadmin [ -r ] [ -c ] [ -v ]
```

A.13.1.3 Description

With the commands listed in this entry, `tmadmin` provides for the inspection and modification of bulletin boards and associated entities in a uniprocessor, multiprocessor, or networked environment. The `TUXCONFIG` and `TUXOFFSET` environment variables are used to determine the location and offset at which the Oracle Tuxedo configuration file is loaded. `tmadmin` supports the following options:

-c

If `tmadmin` is invoked with the `-c` option, it enters configuration mode. The only valid commands are `default`, `echo`, `help`, `quit`, `verbose`, `livtoc`, `crdl`, `lidl`, `dsdl`, `indl`, and `dumplog`. `tmadmin` may be invoked in this mode on any node, including inactive nodes. A node is considered active if `tmadmin` can join the application as an administrative process or client (via a running BBL).

-r

The `-r` option instructs `tmadmin` to enter the bulletin board as a client instead of as the administrator; in other words, it requests read-only access. This option is useful if you want to leave the administrator slot unoccupied.



Note:

If you decide to use this option, however, be aware that you will not get all the information you get by running `tmadmin` without the `-r` option. Specifically, `tmadmin -r` does not report load values for servers running at remote sites.

Only one `tmadmin` process can be the administrator at a time. When the `-r` option is specified by a user other than the Oracle Tuxedo administrator and security is turned on, the user is prompted for a password.

-v

The `-v` option causes `tmadmin` to display the Oracle Tuxedo version number and license number. After printing out the information, `tmadmin` exits. If the `-v` option is entered with either of the other two options, the others are ignored; only the information requested by the `-v` option is displayed.

Normally, `tmadmin` may be run on any active node within an active application. If it is run on an active node that is partitioned or if there is no NLS available for the master node, commands are limited to read-only access to the local bulletin board. These command include `bbls`, `bbparms`, `bbstat`, `default`, `dump`, `dumplog`, `echo`, `help`, `interfaceparms`, `printactiveobject`, `printclient`, `printinterface`, `printfactory`, `printnet`, `printqueue`, `printroute`, `printserver`, `printservice`,

printtrans, printgroup, reconnect, quit, serverparms, serviceparms, and verbose, in addition to the configuration commands. If the partitioned node is the backup node for the MASTER (specified as the second entry on the MASTER parameter in the RESOURCES section of the configuration file), the master command is also available to make this node the MASTER for this part of the partitioned application.

If the application is inactive, tmadmin can be run only on the MASTER processor. In this mode, all of the configuration mode commands are available plus the TLOG commands (crlog, dslog, and inlog) and boot.

Once tmadmin has been invoked, commands may be entered at the prompt (>) according to the following syntax:

```
command [ arguments ]
```

Several commonly occurring arguments can be given defaults via the default command. Commands that accept parameters set via the default command check default to see whether a value has been set. If a value has not been set, an error message is returned.

In a networked or multiprocessor environment, a single bulletin board can be accessed by setting a default machine (the logical machine ID (LMID) as listed in the MACHINES section of the UBBCONFIG file). If the default machine is set to all, all bulletin boards are accessed. If machine is set to DBBL, the distinguished bulletin board is addressed. The default machine is shown as part of the prompt, as in: MASTER>.

If machine is not set via the default command, the DBBL is addressed (the local BBL is used in a SHM configuration).

The machine value for a command can generally be obtained from the default setting (printserver is an example). A caution is required here, however, because some commands (the TLOG commands, for example) act on devices found through TUXCONFIG; a default setting of DBBL or all results in an error. For some commands, such as logstart, you must specify the value of machine on the command line; the value does not appear as an argument to the -m option.

Once set, a default remains in effect until the session is ended, unless changed by another default command. Defaults may be overridden by entering an explicit value on the command line, or unset by entering the value "". The effect of an override lasts for a single instance of the command.

Output from tmadmin commands is paginated according to the pagination command in use (see the description of the paginate subcommand later in this entry).

There are some commands that have either verbose or terse output. The verbose command can be used to set the default output level. However, each command (except boot, shutdown, and config) takes a -v or -t option to turn on verbose or terse output for that command only. When output is printed in terse mode, some information (for example, LMID or GROUP name, service name, or server name) may be truncated. This type of truncation is indicated by a plus sign, +, at the end of the value. The entire value may be seen by reentering the command in verbose mode.

A.13.2 tmadmin Command Related to TMA

Commands may be entered either by their full name or their abbreviation (as given in parentheses), followed by any appropriate arguments. Arguments appearing in square brackets, [], are optional; those in curly braces, {}, indicate a selection from mutually exclusive

options. Note that command line options that do not appear in square brackets need not appear on the command line (that is, they are optional) if the corresponding default has been set via the `default` command. Ellipses following a group of options in curly brackets, `{}`..., indicate that more than one of the options listed may appear on the command line (at least one must appear).

```
changetrace (chtr) [-m machine] [-g groupname] [-i srvid] specification
```

Activate the trace at a specific level for SNA Gateway. The proposed trace specification is:

```
filter-spec: receiver-spec [ :trigger-spec:level-spec]
```

level-spec

Optional. The format is level number or *. If the trace level is not specified, no trace is dumped. If the trace level is specified, only the trace at the level less than or equal to the specified level is dumped. If * is specified, all the trace is dumped.

A new category for TMA trace is added: `snax`

The traditional catalog is written into ULOG, but for other sorts of trace could be dumped into another log file, which is activated by `receiver-spec`, and trace file name prefix could be specified by the environment variable `SNATRACEPREFIX`.

- [Examples](#)

A.13.2.1 Examples

- If trace specification is set as: `snax:ulog:dye:2`
- It indicates the traces at the level lower than 2 are dumped into ULOG. Here `snax` is `filter-spec`, `ulog` is `receiver-spec`, `dye` is `trigger-spec`, and 2 is `level-spec`.
- If trace specification is set as below: `snax:ulog:dye:*`
All levels of trace are dumped into ULOG.
- If trace specification is set as: `snax:utrall:dye:3`
Traces (level 1, 2, and 3) are dumped into specified trace file which prefix is specified by the environment variable: `SNATRACEPREFIX`

If `SNATRACEPREFIX` is not specified, the trace file is: `SNALOG.mmddy`

B

Error Messages

The following messages are logged to the USERLOG file from an Oracle Tuxedo Mainframe Adapter for SNA Gateway (GWSNAX) and the Communications Resource Manager (CRM).

1118:INFO	A BLOCKING timeout has occurred on an outbound request
	DESCRIPTION A blocking time out has occurred on a request to the host.
	ACTION Examine the ULOG for more information concerning the reason for the failure. Check configuration and increase the BLOCKTIME value, if needed.
1121:ERROR	Bad action state
	DESCRIPTION An internal error has been detected processing the action.
	ACTION Contact Oracle Customer Support.
1122:ERROR	Invalid action index
	DESCRIPTION An internal error has been detected, <code>gw_nw_acall</code> has returned a failure.
	ACTION Contact Oracle Customer Support.
1600:ERROR	Invalid action index
	DESCRIPTION An internal error has been detected, <code>gw_nw_connect</code> has returned a failure.
	ACTION Contact Oracle Customer Support.
1602:ERROR	Cannot open SNA connection
	DESCRIPTION Unable to allocate a network context. <code>gw_nw_AllocNwCtx</code> failed.
	ACTION See previously printed ULOG message for cause.
1604:ERROR	Initialization of environment failed
	DESCRIPTION Unable to initialize the conversation environment.
	ACTION See previously printed ULOG message for cause.
1621:ERROR	Bad action state
	DESCRIPTION An internal error has been detected processing the action.
	ACTION Contact Oracle Customer Support.
1622:ERROR	Invalid action index
	DESCRIPTION An internal error has been detected, <code>gw_nw_convsend</code> has returned a failure.
	ACTION Contact Oracle Customer Support.
1625: ERROR	Invalid action index
	DESCRIPTION An internal error has been detected, <code>gw_nw_convsend</code> has returned a failure.
	ACTION Contact Oracle Customer Support.
1625:ERROR	Invalid action index
	DESCRIPTION An internal error has been detected, <code>gw_nw_disconnect</code> has returned a failure.
	ACTION Contact Oracle Customer Support.
1630:ERROR	Unable to send data on sna_conv_idx .

	DESCRIPTION	A network error has occurred while processing a send to the CRM. <code>gw_nw_convsend</code> failed.
	ACTION	Determine if CRM is still active. Look for other ULOG messages that specify a reason for this failure.
1989:ERROR	External encode/decode service returned error: \n\tTUXEDO code ()	
	DESCRIPTION	A Tuxedo error has occurred in the Data Integration portion of the product.
	ACTION	Use the Tuxedo error to resolve the Data Integration problem. If unable to resolve the problem, contact Oracle Customer Support.
1990:ERROR	Cannot create external decode action!	
	DESCRIPTION	An internal error has occurred in the Data Integration portion of the product.
	ACTION	Contact Oracle Customer Support.
1991:ERROR	Cannot create external encode action!	
	DESCRIPTION	An internal error has occurred in the Data Integration portion of the product.
	ACTION	Contact Oracle Customer Support.
1992:ERROR	Invalid action index	
	DESCRIPTION	An internal error has been detected, <code>gw_nw_decode</code> has returned a failure.
	ACTION	Contact Oracle Customer Support.
1993:ERROR	Invalid action index	
	DESCRIPTION	An internal error has been detected, <code>gw_nw_encode</code> has returned a failure.
	ACTION	Contact Oracle Customer Support.
1994:ERROR	Context data incorrect - null context index	
	DESCRIPTION	An internal error has been detected, <code>gw_nw_decode</code> has returned a failure.
	ACTION	Contact Oracle Customer Support.
1995:ERROR	Context data incorrect - null context index	
	DESCRIPTION	An internal error has been detected, <code>gw_nw_encode</code> has returned a failure.
	ACTION	Contact Oracle Customer Support.
2200:ERROR	Cannot malloc remote domain structure	
	DESCRIPTION	An internal error has been detected, <code>gw_nw_init</code> was unable to allocate memory needed.
	ACTION	Contact Oracle Customer Support.
2204:ERROR	Cannot malloc local SNA structur	
	DESCRIPTION	An internal error has been detected, <code>gw_nw_init</code> was unable to allocate memory needed.
	ACTION	Contact Oracle Customer Support.
2205:ERROR	Unable to obtain local snadom information from shm	
	DESCRIPTION	An internal error has been detected, <code>gw_nw_init</code> was unable to get needed information from shared memory.
	ACTION	Contact Oracle Customer Support.
2206:ERROR	Cannot malloc remote service structure	
	DESCRIPTION	An internal error has been detected, <code>gw_nw_init</code> was unable to allocate memory needed.
	ACTION	Contact Oracle Customer Support.

2207:ERROR	Cannot malloc remote SNA structure
	DESCRIPTION An internal error has been detected, <code>gw_nw_init</code> was unable to allocate memory needed.
	ACTION Contact Oracle Customer Support.
2210:ERROR	Can't create listener for this GATEWAY
	DESCRIPTION An internal error has been detected, <code>gw_nw_init</code> was unable to create a listener.
	ACTION Contact Oracle Customer Support.
2211:WARNING	NO SNA conversations waiting for startup
	DESCRIPTION No conversations are waiting at startup.
	ACTION Contact Oracle Customer Support.
2214:ERROR	Unable to set APPC_GATEWAY environment variable
	DESCRIPTION An internal error has been detected, <code>_gp_get_nodename</code> failed.
	ACTION Contact Oracle Customer Support.
2220:ERROR	Memory Allocation Failure
	DESCRIPTION An internal error has been detected, <code>gw_nw_init</code> was unable to allocate memory needed.
	ACTION Contact Oracle Customer Support.
2221:ERROR	Memory Allocation Failure
	DESCRIPTION An internal error has been detected, <code>gw_nw_init</code> was unable to allocate memory needed.
	ACTION Contact Oracle Customer Support.
2222:ERROR	Memory Allocation Failure for RDOM table
	DESCRIPTION An internal error has been detected, <code>gw_nw_init</code> was unable to allocate memory needed.
	ACTION Contact Oracle Customer Support.
2223:ERROR	Memory Allocation Failure for SNALINK table
	DESCRIPTION An internal error has been detected, <code>gw_nw_init</code> was unable to allocate memory needed.
	ACTION Contact Oracle Customer Support.
2224:ERROR	Memory Allocation Failure for CRM table
	DESCRIPTION An internal error has been detected, <code>gw_nw_init</code> was unable to allocate memory needed.
	ACTION Contact Oracle Customer Support.
2225:ERROR	Memory Allocation Failure for SNASTACK table
	DESCRIPTION An internal error has been detected, <code>gw_nw_init</code> was unable to allocate memory needed.
	ACTION Contact Oracle Customer Support.
2226:ERROR	Unable to access SNALINK information from memory
	DESCRIPTION An internal error has been detected, <code>gw_nw_init</code> was unable to get needed information from shared memory.
	ACTION Contact Oracle Customer Support.
2227:ERROR	Unable to access CRM information from memory
	DESCRIPTION An internal error has been detected, <code>gw_nw_init</code> was unable to get needed information from shared memory.
	ACTION Contact Oracle Customer Support.
2228:ERROR	Unable to convert CRM network address for <text>

	DESCRIPTION	An error has occurred while trying to contact the CRM during startup.
	ACTION	Examine the ULOG for more information concerning the reason for the failure. Probably due to a configuration problem.
2229:ERROR	Unable to access SNASTACK information from memory	
	DESCRIPTION	An internal error has been detected, <code>gw_nw_init</code> was unable to get needed information from shared memory.
	ACTION	Contact Oracle Customer Support.
2230:ERROR	Unable to build internal network tables	
	DESCRIPTION	An internal error has been detected, <code>gw_nw_build_nwtbls</code> failed.
	ACTION	Examine the ULOG for more information concerning the reason for the failure. Probably due to a configuration problem.
2231:ERROR	Unable to connect to CRM <text>	
	DESCRIPTION	An error has occurred while trying to contact the CRM during startup.
	ACTION	Examine the ULOG for more information concerning the reason for the failure. Possible reasons are: <ul style="list-style-type: none"> • The CRM server was not configured in the <code>UBBCONFIG</code>. • The <code>//host:port</code> parameters as specified in the <code>DMCONFIG</code> and the CRM command line parameters in the <code>UBBCONFIG</code> do not agree
2232:ERROR	Unable to SIGNON to CRM <text>	
	DESCRIPTION	An error has occurred while signing on to the CRM.
	ACTION	Examine the ULOG for more information concerning the reason for the failure.
2233:ERROR	Unable to send STACK configuration to CRM <text>	
	DESCRIPTION	An error has occurred while sending the stack configuration to the CRM.
	ACTION	Examine the ULOG for more information concerning the reason for the failure.
2234:ERROR	Unable to send SNALINK configuration to CRM <text>	
	DESCRIPTION	An error has occurred while sending the link configuration to the CRM.
	ACTION	Examine the ULOG for more information concerning the reason for the failure.
2235:ERROR	Unable to send START message to CRM	
	DESCRIPTION	An error has occurred while sending the START command to the CRM.
	ACTION	Examine the ULOG for more information concerning the reason for the failure.
2236:ERROR	Unable to send SHUTDOWN message to CRM <text>	
	DESCRIPTION	An error has occurred while sending the SHUTDOWN command to the CRM.
	ACTION	Examine the ULOG for more information concerning the reason for the failure.
2237:ERROR	STATE ERROR occurred during initialization, SHUTTING DOWN	
	DESCRIPTION	An internal state error has occurred start up.
	ACTION	Contact Oracle Customer Support.

2238:ERROR	Unknown message type received from CRM during initialization
	DESCRIPTION An unknown message has been received from CRM.
	ACTION Contact Oracle Customer Support.
2239:ERROR	GATEWAY FAILS to complete connection to the CRM during initialization
	DESCRIPTION Connection to CRM is incomplete.
	ACTION Examine the ULOG for more information concerning the reason for the failure.
2240:ERROR	Unable to send SHUTDOWN message to CRM <text>
	DESCRIPTION GWSNAX is unable to send the SHUTDOWN command to CRM.
	ACTION Examine the ULOG for more information concerning the reason for the failure. The CRM process has probably already terminated.
2241:ERROR NA	Unable to access codepage table [<text>], reason= <text> (<number>)
	DESCRIPTION Oracle Tuxedo Mainframe Adapter for SNA Gateway is unable to find the file for the listed codepage for the reason specified. If reason is "Error 0", the () value is the return code from the codepage read routine. 2 - Can't read from file 3 - - Can't allocate memory 4 - Null/invalid pointer 5 - Invalid parameter 10 - Syntax error 11 - Unsupported format version 12 - Unsupported item
NA	ACTION Examine the DMCONFIG to determine if the CODEPAGE for a remote domain is specified incorrectly. Correct and retry.
2247:INFO NA	Codepage for Remote Domain : <Remote Domain Name> Codepage
	DESCRIPTION Examine the ULOG for more information regarding the Codepage used for the remote domain.
	ACTION No action needs to be taken.
2302:ERROR	Unable to complete initialization with CRM
	DESCRIPTION Connection to CRM is incomplete.
	ACTION Examine the ULOG for more information concerning the reason for the failure.
2509:ERROR	Unable to obtain new convid id for this request
	DESCRIPTION An internal error has been detected, gw_nw_getnxt_cd has returned a failure.
	ACTION Contact Oracle Customer Support.
2561:ERROR	Action data incorrect - null action index
	DESCRIPTION An internal error has been detected in gw_nw_recv.
	ACTION Contact Oracle Customer Support.
2601:ERROR	Unable to send data on sna_conv_idx <number>
	DESCRIPTION An error has occurred while sending reply data (gw_nw_reply) to the CRM.
	ACTION Examine the ULOG for more information concerning the reason for the failure.
2602:ERROR	Null send data buffer
	DESCRIPTION An error has occurred while sending reply data (gw_nw_reply) to the CRM.

	ACTION	Examine the ULOG for more information concerning the reason for the failure.
2800:ERROR	Invalid action index	
	DESCRIPTION	An internal error has been detected in <code>gw_nw_clr_convld</code> .
	ACTION	Contact Oracle Customer Support.
2801:ERROR	Invalid conversation id	
	DESCRIPTION	An internal error has been detected in <code>gw_nw_clr_convld</code> .
	ACTION	Contact Oracle Customer Support.
2802:ERROR	Invalid action index	
	DESCRIPTION	An internal error has been detected in <code>gw_nw_getnxt_cd</code> .
	ACTION	Contact Oracle Customer Support.
2805:ERROR	Unable to obtain remote domain info from shared memory	
	DESCRIPTION	An internal error has been detected, <code>gw_nw_init_env</code> was unable to get needed information from shared memory.
	ACTION	Contact Oracle Customer Support.
2806:ERROR	Unable to obtain snadom info from rdsm	
	DESCRIPTION	An internal error has been detected, <code>gw_nw_init_env</code> was unable to get needed information from shared memory.
	ACTION	Contact Oracle Customer Support.
2808:ERROR	Cannot realloc sna structures	
	DESCRIPTION	An internal error has been detected, <code>gw_nw_AllocNwCtx</code> was unable to allocate memory needed.
	ACTION	Contact Oracle Customer Support.
2816:ERROR	Memory allocation failure	
	DESCRIPTION	An internal error has been detected, <code>gw_nw_empty_sndbuf</code> was unable to allocate memory needed.
	ACTION	Contact Oracle Customer Support.
2847:ERROR	SECURITY: Unable to get remote username from shared memory using local principal name <text>	
	DESCRIPTION	A security error has occurred, the user is not authorized to access the remote host.
	ACTION	Contact your local security administrator.
2848:ERROR	SECURITY: Unable to get local principal name from shared memory using appkey	
	DESCRIPTION	A security error has occurred, the user is not authorized to access the remote host.
	ACTION	Contact your local security administrator.
2849:ERROR	SECURITY: Unable to extract password from shared memory	
	DESCRIPTION	A security error has occurred, the user is not authorized to access the remote host.
	ACTION	Contact your local security administrator.
2850:ERROR	SECURITY: Unable to get security schedule	
	DESCRIPTION	A security error has occurred, the user is not authorized to access the remote host.
	ACTION	Contact your local security administrator.
2853:ERROR	Invalid action index	
	DESCRIPTION	An internal error has been detected in <code>gw_nw_mk_error</code> .
	ACTION	Contact Oracle Customer Support.
2854:ERROR	Context data incorrect - null context index for action <number>	

	DESCRIPTION	An internal error has been detected in gw_nw_mk_error.
	ACTION	Contact Oracle Customer Support.
2860:ERROR	SECURITY: Unable to get remote user from shared memory	
	DESCRIPTION	A security error has occurred, the remote user is not authorized to access the local Tuxedo host.
	ACTION	Contact your local security administrator.
2935:ERROR	Too many unrecoverable errors occurred - deleting action	
	DESCRIPTION	An error has occurred in gw_nw_mk_error.
	ACTION	Examine the ULOG for more information concerning the reason for the failure.
2938:ERROR	SECURITY: Unable to get appkey for this outbound request	
	DESCRIPTION	A security error has occurred, the user is not authorized to access the remote host.
	ACTION	Contact your local security administrator.
2939:ERROR	Deleting the action with the invalid context, <number>	
	DESCRIPTION	An internal error has been detected in gw_nw_snd failure.
	ACTION	Contact Oracle Customer Support.
3028:ERROR	Can't find transaction node for Network TMS event!	
	DESCRIPTION	An internal error has been detected in gw_nw_TransCommon.
	ACTION	Contact Oracle Customer Support.
3029:WARNING	Shutting down GWSNAX!	
	DESCRIPTION	A severe error such as a protocol error or memory corruption has occurred and has already been logged. The GWSNAX process will shutdown.
	ACTION	Examine the ULOG for more information concerning the reason for the failure.
3030:ERROR	Unknown input transactional event (<number>)! 	
	DESCRIPTION	An internal error has been detected in gw_nw_TransCvtMsgEvt2TranEvt.
	ACTION	Contact Oracle Customer Support.
3031:ERROR	Invalid input object pointer!	
	DESCRIPTION	An internal error has been detected in gw_nw_BindStateObject.
	ACTION	Contact Oracle Customer Support.
3032:ERROR	Invalid input object initial state!	
	DESCRIPTION	An internal error has been detected in gw_nw_BindStateObject.
	ACTION	Contact Oracle Customer Support.
3033:ERROR	Invalid input object type!	
	DESCRIPTION	An internal error has been detected in gw_nw_BindStateObject.
	ACTION	Contact Oracle Customer Support.
3034:ERROR	Invalid input state object event(<text>)! 	
	DESCRIPTION	An internal error has been detected in gw_nw_StateObjectTestEvent.
	ACTION	Contact Oracle Customer Support.
3035:ERROR	Event <text> is not allowed for state <text> !	

	DESCRIPTION	An internal error has been detected in gw_nw_StateObjectTestEvent.
	ACTION	Contact Oracle Customer Support.
3036:ERROR	Invalid object event test metric with value <number> for <text>!	
	DESCRIPTION	An internal error has been detected in gw_nw_StateObjectTestEvent.
	ACTION	Contact Oracle Customer Support.
3037:ERROR	Invalid input trace object pointer!	
	DESCRIPTION	An internal error has been detected in gw_nw_InitEventTrace.
	ACTION	Contact Oracle Customer Support.
3038:ERROR	Invalid input trace object pointer!	
	DESCRIPTION	An internal error has been detected in gw_nw_AddEventTrace.
	ACTION	Contact Oracle Customer Support.
3039:ERROR	Input event <number> out of range <number> !	
	DESCRIPTION	An internal error has been detected in gw_nw_AddEventTrace.
	ACTION	Contact Oracle Customer Support.
3040:ERROR	Invalid input trace object pointer!	
	DESCRIPTION	An internal error has been detected in gw_nw_QueryEventTrace.
	ACTION	Contact Oracle Customer Support.
3041:ERROR	Invalid input trace object pointer!	
	DESCRIPTION	An internal error has been detected in gw_nw_PrintEventTrace.
	ACTION	Contact Oracle Customer Support.
3042:ERROR	Invalid input event identification(<number>)!	
	DESCRIPTION	An internal error has been detected in gw_nw_EventToString.
	ACTION	Contact Oracle Customer Support.
3044:WARNING	Transaction table already exists!	
	DESCRIPTION	An attempt has been made to create the node table and it already exists.
	ACTION	No action needs to be taken.
3045:ERROR	Invalid input parameters!	
	DESCRIPTION	An internal error has been detected in gw_nw_AssociateTranByTXID.
	ACTION	Contact Oracle Customer Support.
3046:ERROR	Requires CRM id to create transaction!	
	DESCRIPTION	An internal error has been detected in gw_nw_AssociateTranByTXID.
	ACTION	Contact Oracle Customer Support.
3047:ERROR	Cannot associate with network transaction!	
	DESCRIPTION	An internal error has been detected in gw_nw_AssociateTranByTXID.
	ACTION	Contact Oracle Customer Support.
3048:ERROR	Invalid input transaction context!	

	DESCRIPTION	An internal error has been detected in gw_nw_AssociateTranByTCTXT.
	ACTION	Contact Oracle Customer Support
3049:ERROR	Invalid input transaction event(<number>)! DESCRIPTION	
		An internal error has been detected in gw_nw_AssociateTranByTCTXT.
	ACTION	Contact Oracle Customer Support.
3050:ERROR	Invalid input transaction context pointer! DESCRIPTION	
		An internal error has been detected in gw_nw_TranCreateTCTXT.
	ACTION	Contact Oracle Customer Support.
3051:ERROR	Exceeded local transaction limit (<number>)! DESCRIPTION	
		Number of concurrent transactions using this domain gateway exceeded the configured limit.
	ACTION	Examine the DMCONFIG file for the MAXTRAN parameter, this number is the limit. If MAXTRAN is not specified, then examine the ubbconfig file for the MAXGTT parameter. In the later case MAXGTT is the limit. Modify the limit if necessary.
3052:ERROR	Can not create shared memory nettxid! DESCRIPTION	
		An internal error has been detected in gw_nw_TranCreateTCTXT.
	ACTION	Contact Oracle Customer Support.
3053:ERROR	Create transaction tree node failed! DESCRIPTION	
		gw_nw_TranCreateTCTXT failed to create the transaction tree node object. This can be caused by reaching the configured transaction limit or by a memory allocation failure.
	ACTION	Examine the ULOG for more information concerning the reason for the failure. If the cause is the limit being reached, update the configuration. If the cause is a memory allocation failure, update the system and/or the system configuration.
3054:ERROR	Invalid input transaction handle! DESCRIPTION	
		An internal error has been detected in gw_nw_TranGetTCTXT.
	ACTION	Contact Oracle Customer Support.
3055:ERROR	Invalid input transaction handle! DESCRIPTION	
		An internal error has been detected in gw_nw_TranGetTXID.
	ACTION	Contact Oracle Customer Support.
3056:ERROR	Invalid input parameter! DESCRIPTION	
		An internal error has been detected in gw_nw_TranProcessEvent.
	ACTION	Contact Oracle Customer Support.
3057:ERROR	Unable to calculate transaction handle! DESCRIPTION	
		An internal error has been detected in gw_nw_TranProcessEvent.
	ACTION	Contact Oracle Customer Support.
3058:ERROR	Input Gateway event not transactional! DESCRIPTION	
		The service request inbound from a remote host is transactional but the Oracle Tuxedo Mainframe Adapter for SNA Gateway is not configured as a transactional gateway.

	ACTION	The MAXSYNCLVL specified in the DMCONFIG must be 2 if transactional requests are to be processed.
3059:ERROR	Transaction does not exist!	
	DESCRIPTION	An internal error has been detected in gw_nw_TransProcessEvent.
	ACTION	Contact Oracle Customer Support.
3060:ERROR	Invalid input transaction handle!	
	DESCRIPTION	An internal error has been detected in gw_nw_TransProcessEvent.
	ACTION	Contact Oracle Customer Support.
3061:ERROR	Invalid Gateway action identification!	
	DESCRIPTION	An internal error has been detected in gw_nw_TransSetUpExtTmsEvent.
	ACTION	Contact Oracle Customer Support.
3062:ERROR	Decoding input transaction message failed!	
	DESCRIPTION	An internal error has been detected in gw_nw_TransSetUpExtTmsEvent.
	ACTION	Contact Oracle Customer Support.
3063:ERROR	CRM verb <number> is not for TMS event!	
	DESCRIPTION	An internal error has been detected in gw_nw_TransSetUpExtTmsEvent.
	ACTION	Contact Oracle Customer Support.
3064:ERROR	Invalid input transaction context!	
	DESCRIPTION	An internal error has been detected in gw_nw_TransRelease.
	ACTION	Contact Oracle Customer Support.
3065:ERROR	Transaction for context(<number>) not found!	
	DESCRIPTION	The specified context for the transaction to be released could not be found by gw_nw_TransRelease.
	ACTION	Examine the ULOG for more information concerning the reason for the failure.
3066:ERROR	Cannot disassociate with network transaction!	
	DESCRIPTION	An error has been returned by gw_tx_end.
	ACTION	Examine the ULOG for more information concerning the reason for the failure.
3067:ERROR	Cannot set remote domain for transaction!	
	DESCRIPTION	An error has been returned by gw_tx_set_rdom. Several possibilities exist for this failure. <ul style="list-style-type: none"> • The process is unable to obtain the shared memory lock. • The configured limit for the number of remote domains that can be involved in the same transaction has been reached. This limit is specified in the MAXRDTRAN in the DMCONFIG.
	ACTION	Examine the DMCONFIG and compare with the number of remote domains which might become involved in the same transaction. Update the DMCONFIG if necessary. Also examine the ULOG for more information concerning possible reasons for the failure.
3068:ERROR	Invalid input transaction handle!	

	DESCRIPTION	An internal error has been detected in gw_nw_TranGetNodePtr.
	ACTION	Contact Oracle Customer Support.
3069:ERROR	Transaction handle not in use!	
	DESCRIPTION	An internal error has been detected in gw_nw_TranGetNodePtr.
	ACTION	Contact Oracle Customer Support.
3070:ERROR	Transaction processing not yet initialize!	
	DESCRIPTION	An internal error has been detected in gw_nw_TranGetNodePtr.
	ACTION	Contact Oracle Customer Support.
3071:ERROR	Defined transaction limit reached!	
	DESCRIPTION	The number of concurrent transactions using this domain gateway has exceeded the limit.
	ACTION	Examine the DMCONFIG file for the MAXTRAN parameter, this number is the limit. If MAXTRAN is not specified, then examine the ubbconfig file for the MAXGTT parameter. In the later case MAXGTT is the limit. Modify the limit if necessary.
3072:ERROR	Allocate CRM branch structure failed!	
	DESCRIPTION	An internal error has been detected in gw_nw_gw_nw_AllocateNode.
	ACTION	Contact Oracle Customer Support.
3073:ERROR	Unable to bind outbound node state object!	
	DESCRIPTION	A failure was returned by gw_nw_BindStateObject.
	ACTION	Examine the ULOG for more information concerning the reason for the failure.
3074:ERROR	Unable to bind inbound node state object!	
	DESCRIPTION	A failure was returned by gw_nw_BindStateObject.
	ACTION	Examine the ULOG for more information concerning the reason for the failure.
3075:ERROR	Unable to bind recover node state object!	
	DESCRIPTION	A failure was returned by gw_nw_BindStateObject.
	ACTION	Examine the ULOG for more information concerning the reason for the failure.
3076:ERROR	Invalid event to create transaction tree node!	
	DESCRIPTION	An internal error has been detected in _gw_nw_AllocateNode.
	ACTION	Contact Oracle Customer Support.
3077:ERROR	Memory allocation failure!	
	DESCRIPTION	An internal error has been detected in gw_nw_AllocBranch.
	ACTION	Contact Oracle Customer Support.
3078:ERROR	Event(<text>) is not allowed to create CRM branch!	
	DESCRIPTION	An internal error has been detected in gw_nw_AllocBranch.
	ACTION	Contact Oracle Customer Support.
3079:ERROR	Free more branch than transaction tree node has!	
	DESCRIPTION	An internal error has been detected in gw_nw_FreeBranch.
	ACTION	Contact Oracle Customer Support.
3080:ERROR	Protocol violation, receive event(<text>) at state(<text>)! 	

	DESCRIPTION	An internal error has been detected in gw_nw_NodeMethod_3.
	ACTION	Contact Oracle Customer Support.
3081:ERROR	Invalid Gateway action, receive event(<text>) at state(<text>)!	
	DESCRIPTION	An internal error has been detected in gw_nw_NodeMethod_4.
	ACTION	Contact Oracle Customer Support.
3082:ERROR	Invalid Gateway action, receive event(<text>) at state(<text>)!	
	DESCRIPTION	An internal error has been detected in gw_nw_NodeMethod_5.
	ACTION	Contact Oracle Customer Support.
3083:ERROR	Branch in a bad state(<text>) to issue prepare!	
	DESCRIPTION	An internal error has been detected in gw_nw_NodeMethod_5.
	ACTION	Contact Oracle Customer Support.
3084:WARNING	Inconsistent branch state for prepare(<number>,<number>)!	
	DESCRIPTION	The transaction branch is in an inconsistent state for issuing a prepare. It will be assumed that they are all prepared.
	ACTION	No action needs to be taken.
3085:INFO	Transaction node is deallocated because of ABEND!	
	DESCRIPTION	Informational message. Oracle Tuxedo Mainframe Adapter for SNA Gateway detected an ABEND condition or received an ABEND event from the CRM which caused the node object to be destroyed.
	ACTION	No action needs to be taken.
3086:ERROR	Invalid Gateway action, receive event(<text>) at state(<text>)!	
	DESCRIPTION	An internal error has been detected in gw_nw_NodeMethod_7.
	ACTION	Contact Oracle Customer Support.
3087:ERROR	Branch in a bad state(<text>) to issue prepare!	
	DESCRIPTION	An internal error has been detected in gw_nw_NodeMethod_7.
	ACTION	Contact Oracle Customer Support.
3088:WARNING	Inconsistent branch state for rollback(<number>,<number>)!	
	DESCRIPTION	The transaction branch was in an inconsistent state for issuing a rollback. It will be assumed that they are all rolled back.
	ACTION	No action needs to be taken.
3089:WARNING	Inconsistent branch accounting for rollback(<number>,<number>)!	
	DESCRIPTION	The transaction branch accounting was inconsistent while issuing a rollback.
	ACTION	No action needs to be taken.
3090:WARNING	Incorrect branch accounting for rollback(<number>)!	
	DESCRIPTION	The transaction branch accounting was incorrect while issuing a rollback.
	ACTION	No action needs to be taken.
3091:ERROR	Invalid Gateway action, receive event(<text>) at state(<text>)!	
	DESCRIPTION	An internal error has been detected in gw_nw_NodeMethod_8.

	ACTION	
3092:ERROR	Invalid Gateway action, receive event(<text>) at state(<text>)! DESCRIPTION	An internal error has been detected in gw_nw_NodeMethod_9.
	ACTION	Contact Oracle Customer Support
3093:ERROR	Invalid Gateway action, receive event(<text >) at state(<text>)! DESCRIPTION	An internal error has been detected in gw_nw_NodeMethod_10.
	ACTION	Contact Oracle Customer Support.
3094:ERROR	Invalid Gateway action, receive event(<text>) at state(<text>)! DESCRIPTION	An internal error has been detected in gw_nw_NodeMethod_11.
	ACTION	Contact Oracle Customer Support.
3095:ERROR	Invalid Gateway action, receive event(<text>) at state(<text>)! DESCRIPTION	An internal error has been detected in gw_nw_NodeMethod_12.
	ACTION	Contact Oracle Customer Support.
3096:ERROR	Invalid Gateway action, receive event(<text>) at state(<text>)! DESCRIPTION	An internal error has been detected in gw_nw_NodeMethod_13.
	ACTION	Contact Oracle Customer Support.
3097:ERROR	Invalid Gateway action, receive event(<text>) at state(<text>)! DESCRIPTION	An internal error has been detected in gw_nw_NodeMethod_14.
	ACTION	Contact Oracle Customer Support.
3098:ERROR	Invalid Gateway action, receive event(<text>) at state(<text>)! DESCRIPTION	An internal error has been detected in gw_nw_NodeMethod_15.
	ACTION	Contact Oracle Customer Support.
3099:ERROR	Retired node method! DESCRIPTION	An internal error has been detected in gw_nw_NodeMethod_Retired.
	ACTION	Contact Oracle Customer Support.
3100:ERROR	Invalid Gateway action, receive event(<text>) at state(<text>)! DESCRIPTION	An internal error has been detected in gw_nw_NodeMethod_16.
	ACTION	Contact Oracle Customer Support.
3101:ERROR	Invalid Gateway action, receive event(<text>) at state(<text>)! DESCRIPTION	An internal error has been detected in gw_nw_NodeMethod_17.
	ACTION	Contact Oracle Customer Support.
3102:ERROR	Invalid Gateway action, receive event(<text>) at state(<text>)! DESCRIPTION	An internal error has been detected in gw_nw_NodeMethod_18.
	ACTION	Contact Oracle Customer Support.
3103:ERROR	Invalid Gateway action, receive event(<text>) at state(<text>)! DESCRIPTION	An internal error has been detected in gw_nw_NodeMethod_19.
	ACTION	Contact Oracle Customer Support.

3104:ERROR	Invalid Gateway action, receive event(<text>) at state(<text>)! DESCRIPTION	An internal error has been detected in gw_nw_NodeMethod_20.
	ACTION	Contact Oracle Customer Support.
3105:ERROR	Invalid Gateway action, receive event(<text>) at state(<text>)! DESCRIPTION	An internal error has been detected in gw_nw_NodeMethod_21.
	ACTION	Contact Oracle Customer Support.
3106:ERROR	Invalid Gateway action, receive event(<text>) at state(<text>)! DESCRIPTION	An internal error has been detected in gw_nw_NodeMethod_22.
	ACTION	Contact Oracle Customer Support.
3107:ERROR	Invalid Gateway action, receive event(<text>) at state(<text>)! DESCRIPTION	An internal error has been detected in gw_nw_NodeMethod_23.
	ACTION	Contact Oracle Customer Support.
3108:ERROR	Invalid Gateway action, receive event(<text>) at state(<text>)! DESCRIPTION	An internal error has been detected in gw_nw_NodeMethod_24.
	ACTION	Contact Oracle Customer Support.
3109:ERROR	Invalid Gateway action, receive event(<text>) at state(<text>)! DESCRIPTION	An internal error has been detected in gw_nw_NodeMethod_25.
	ACTION	Contact Oracle Customer Support.
3110:ERROR	Invalid Gateway action, receive event(<text>) at state(<text>)! DESCRIPTION	An internal error has been detected in gw_nw_NodeMethod_26.
	ACTION	Contact Oracle Customer Support.
3111:ERROR	Invalid Gateway action, receive event(<text>) at state(<text>)! DESCRIPTION	An internal error has been detected in gw_nw_NodeMethod_27.
	ACTION	Contact Oracle Customer Support.
3112:ERROR	Invalid Gateway action, receive event(<text>) at state(<text>)! DESCRIPTION	An internal error has been detected in gw_nw_NodeMethod_28.
	ACTION	Contact Oracle Customer Support.
3113:ERROR	Invalid Gateway action, receive event(<text>) at state(<text>)! DESCRIPTION	An internal error has been detected in gw_nw_NodeMethod_29.
	ACTION	Contact Oracle Customer Support.
3114:ERROR	Invalid Gateway action, receive event(<text>) at state(<text>)! DESCRIPTION	An internal error has been detected in gw_nw_NodeMethod_30.
	ACTION	Contact Oracle Customer Support.
3115:ERROR	Invalid Gateway action, receive event(<text>) at state(<text>)! DESCRIPTION	An internal error has been detected in gw_nw_NodeMethod_31.
	ACTION	Contact Oracle Customer Support.
3116:ERROR	Invalid Gateway action, receive event(<text>) at state(<text>)!	

	DESCRIPTION	An internal error has been detected in gw_nw_NodeMethod_32.
	ACTION	Contact Oracle Customer Support.
3117:ERROR	Invalid Gateway action, receive event(<text>) at state(<text>)!	
	DESCRIPTION	An internal error has been detected in gw_nw_NodeMethod_33.
	ACTION	Contact Oracle Customer Support.
3118:ERROR	Invalid Gateway action, receive event(<text>) at state(<text>)!	
	DESCRIPTION	An internal error has been detected in gw_nw_NodeMethod_39.
	ACTION	Contact Oracle Customer Support.
3119:ERROR	Invalid Gateway action, receive event(<text>) at state(<text>)!	
	DESCRIPTION	An internal error has been detected in gw_nw_NodeMethod_40.
	ACTION	Contact Oracle Customer Support.
3120:ERROR	Invalid Gateway action, receive event(<text>) at state(<text>)!	
	DESCRIPTION	An internal error has been detected in gw_nw_NodeMethod_42.
	ACTION	Contact Oracle Customer Support.
3121:ERROR	Invalid Gateway action, receive event(<text>) at state(<text>)!	
	DESCRIPTION	An internal error has been detected in gw_nw_NodeMethod_43.
	ACTION	Contact Oracle Customer Support.
3122:ERROR	Encoding transaction message for verb(<number>) failed!	
	DESCRIPTION	A failure was returned by CRM_tx_msg.
	ACTION	Contact Oracle Customer Support.
3123:ERROR	CRM(<number>) in bad state(<text>) to issue verb(<number>)!	
	DESCRIPTION	An internal error has been detected in gw_nw_NodeSendToBranch.
	ACTION	Contact Oracle Customer Support.
3124:ERROR	Send verb <number> to CRM failed!	
	DESCRIPTION	A failure has been detected while sending the specified verb to the CRM.
	ACTION	Examine the ULOG for more information concerning the reason for the failure. Determine if the CRM process is still active.
3125:ERROR	Encoding transactional message for verb(<number>) failed!	
	DESCRIPTION	A failure was returned by CRM_tx_msg.
	ACTION	Contact Oracle Customer Support.
3126:ERROR	CRM(<number>) in bad state(<text>) to issue verb(<number>)!	
	DESCRIPTION	An internal error has been detected in gw_nw_NodeSendToSubordinate.
	ACTION	Contact Oracle Customer Support.
3127:ERROR	Send verb(<number>) to CRM failed!	
	DESCRIPTION	A failure has been detected while sending the specified verb to the CRM.
	ACTION	Examine the ULOG for more information concerning the reason for the failure. Determine if the CRM process is still active.
3128:ERROR	Encoding transactional message for verb(<number>) failed!	

	DESCRIPTION	A failure was returned by CRM_tx_msg
	ACTION	Contact Oracle Customer Support.
3129:ERROR	CRM(<number>) in bad state(<text>) to issue verb(<number>)!	
	DESCRIPTION	An internal error has been detected in gw_nw_NodeSendToSubordinate.
	ACTION	Contact Oracle Customer Support.
3130:ERROR	Send verb(<number>) to CRM failed!	
	DESCRIPTION	A failure has been detected while sending the specified verb to the CRM.
	ACTION	Examine the ULOG for more information concerning the reason for the failure. Determine if the CRM process is still active.
3131:ERROR	Protocol violation, branch id[<number>] at state(<text>) receive event(<text>)!	
	DESCRIPTION	An internal error has been detected in gw_nw_BranchMethod_1.
	ACTION	
3132:ERROR	Memory allocation error!	
	DESCRIPTION	An internal error has been detected in gw_nw_TransRestart.
	ACTION	Contact Oracle Customer Support.
3133:ERROR	Unable to get shared memory semaphore lock!	
	DESCRIPTION	An internal error has been detected in gw_nw_TransRestart.
	ACTION	Contact Oracle Customer Support.
3134:ERROR	Unable to find transaction on GTT!	
	DESCRIPTION	An internal error has been detected in gw_nw_TransRestart.
	ACTION	Contact Oracle Customer Support.
3135:ERROR	SNAX group is not participating the transaction!	
	DESCRIPTION	An internal error has been detected in gw_nw_TransRestart.
	ACTION	Contact Oracle Customer Support.
3136:ERROR	Unable to setup transaction recovery!	
	DESCRIPTION	A failure was returned by gw_nw_TransRecover.
	ACTION	Contact Oracle Customer Support.
3137:ERROR	Unable to setup transaction recovery!	
	DESCRIPTION	A failure was returned by gw_nw_TransRecover.
	ACTION	Contact Oracle Customer Support.
3138:ERROR	Unable to create recovery action!	
	DESCRIPTION	A failure was returned by gw_new_action.
	ACTION	Contact Oracle Customer Support.
3139:ERROR	Unable to create recovery action!	
	DESCRIPTION	A failure was returned by gw_chld_action.
	ACTION	Contact Oracle Customer Support.
3140:ERROR	Unable to setup transaction recovery!	
	DESCRIPTION	A failure was returned by gw_nw_TransRecover.
	ACTION	Contact Oracle Customer Support.
3141: ERROR	Unable to create recovery action!	

	DESCRIPTION	A failure was returned by <code>gw_new_action</code> .
	ACTION	Contact Oracle Customer Support.
3142:ERROR	Unable to create recovery action!	
	DESCRIPTION	A failure was returned by <code>gw_chld_action</code> .
	ACTION	Contact Oracle Customer Support.
3143:ERROR	Unable to setup transaction recovery!	
	DESCRIPTION	A failure was returned by <code>gw_nw_TransRecover</code> .
	ACTION	Contact Oracle Customer Support.
3144:ERROR	Unable to setup transaction recovery!	
	DESCRIPTION	A failure was returned by <code>gw_nw_TransRecover</code> .
	ACTION	Contact Oracle Customer Support.
3145:ERROR	Unable to release shared memory lock!	
	DESCRIPTION	A failure was returned by <code>_gw_shmunlock</code> .
	ACTION	Contact Oracle Customer Support.
3146:ERROR	Unable to find CRM for superior remote domain(<text>)!	
	DESCRIPTION	Failed to find the remote domain definition in the configuration in order to get the CRM identification.
	ACTION	Examine the DMCONFIG for changes between the boots that would modify or remove the remote definition. If no such change has occurred, contact Oracle Customer Support.
3147:ERROR	Could not find Remote Domain(<text>) to do recovery!	
	DESCRIPTION	An internal error has been detected in <code>gw_nw_TransRecover</code> .
	ACTION	Contact Oracle Customer Support.
3148:ERROR	Could not find Remote Domain to do recovery!	
	DESCRIPTION	An internal error has been detected in <code>gw_nw_TransRecover</code> .
	ACTION	Contact Oracle Customer Support.
3149:ERROR	Could not create transaction tree structure!	
	DESCRIPTION	Unable to allocate a transaction tree node structure to represent the transaction recovery object.
	ACTION	Examine the configuration to determine if there has been a change of MAXTRAN in DMCONFIG and/or a change of MAXGTT in UBBCONFIG. This condition may occur if there is a decrease in the number of transactions allowed between boots while many transactions must be recovered. If no changes have been made, contact Oracle Customer Support.
3150:ERROR	Unable to create CRM branch structure!	
	DESCRIPTION	A failure was returned by <code>_gw_nw_AllocBranch</code> .
	ACTION	Contact Oracle Customer Support.
3151:ERROR	Unable to create CRM branch structure!	
	DESCRIPTION	A failure was returned by <code>_gw_nw_AllocBranch</code> .
	ACTION	Contact Oracle Customer Support.
3152:ERROR	Invalid Gateway action, receive event(<text>) at state(<text>)!	
	DESCRIPTION	An internal error has been detected in <code>gw_nw_NodeMethod_44</code> .
	ACTION	Contact Oracle Customer Support.
3153:ERROR	Invalid Gateway action, receive event(<text>) at state(<text>)!	

	DESCRIPTION	An internal error has been detected in gw_nw_NodeMethod_45.
	ACTION	Contact Oracle Customer Support.
3154:ERROR	Invalid Gateway action, receive event(<text>) at state(<text>)! DESCRIPTION	
	DESCRIPTION	An internal error has been detected in gw_nw_NodeMethod_46.
	ACTION	Contact Oracle Customer Support.
3155:ERROR	Invalid transaction context(<number>) from CRM! DESCRIPTION	
	DESCRIPTION	An internal error has been detected in gw_nw_TransSetupExtTmsEvent.
	ACTION	Contact Oracle Customer Support.
3156:WARNING	Undefined instrumentation register <number> specification, ignored! DESCRIPTION	
	DESCRIPTION	An internal error has been detected in gw_nw_SetUpInstrumentation.
	ACTION	Contact Oracle Customer Support.
3157:INFO	CONNECTED TO CRM DESCRIPTION	
	DESCRIPTION	The Oracle Tuxedo Mainframe Adapter for SNA Gateway process has successfully started the Gateway connection to the CRM process.
	ACTION	No action needs to be taken.
3158:WARNING	CRM Link <text> Inoperable DESCRIPTION	
	DESCRIPTION	The specified link did not start up correctly.
	ACTION	Examine the ULOG for more information concerning the reason for the failure.
3159:ERROR	INVALID STARTTYPE (<text>) - calling shutdown DESCRIPTION	
	DESCRIPTION	The STARTTYPE specified in the DMCONFIG must be AUTO or COLD.
	ACTION	Correct the value specified in the DMCONFIG and restart.
3500:ERROR	Unable to find remote service definition link DESCRIPTION	
	DESCRIPTION	An internal error has been detected in gw_nw_decode.
	ACTION	Contact Oracle Customer Support.
3501:ERROR	Unable to find local service definition link DESCRIPTION	
	DESCRIPTION	An internal error has been detected in gw_nw_decode.
	ACTION	Contact Oracle Customer Support.
3502:ERROR	Buffer type (<text>,<text>) not defined in the buffer type switch! DESCRIPTION	
	DESCRIPTION	The buffer type associated with the inbound message is not specified in the type switch.
	ACTION	Examine the application and the configuration to determine which buffer type is being used. Oracle Tuxedo Mainframe Adapter for SNA Gateway only supports standard Tuxedo buffer types over the domain gateway. If the buffer type is VIEW, VIEW32, then also examine the environment variables VIEWDIR, VIEWFILES, VIEW32DIR, or VIEW32FILES.
3503:ERROR	Failed to tmalloc() STRING buffer of size <number> DESCRIPTION	
	DESCRIPTION	No buffer of type STRING was returned by tmalloc for the size specified.
	ACTION	If the size specified is 1, then a length of zero was received. Check the application to determine if it has an error, otherwise contact Oracle Customer Support.
3504:ERROR	Failed to tmalloc() CARRAY/X_OCTET buffer of size <number>	

	DESCRIPTION	No buffer of type CARRAY or X_OCTET was returned by tmalloc for the size specified.
	ACTION	If the size specified is 1, then a length of zero was received. Check the application to determine if it has an error, otherwise contact Oracle Customer Support.
3505:ERROR	Input buffer too big (<number>)! DESCRIPTION	The buffer received of the size specified, exceeded the maximum allowed of 32767.
	ACTION	Check the application to determine if it has an error, otherwise contact Oracle Customer Support.
3506:ERROR	Missing subtype specification for <text> DESCRIPTION	The buffer type specified of VIEW, VIEW32, X_C_TYPE, or X_COMMON requires that a subtype be specified.
	ACTION	Examine the application and the DMCONFIG and correct the problem.
3507:ERROR	Malloc temporary buffer of size(<number>) failed! DESCRIPTION	An internal error has been detected in gw_nw_decode.
	ACTION	Contact Oracle Customer Support.
3508:ERROR	Convert buffer type(<text>,<text>) failed, convert failure! DESCRIPTION	The conversion of the buffer received from the remote host was unsuccessful.
	ACTION	Examine the application to determine if it has an error, examine the DMCONFIG definition for any errors in the buffer type specification. If no errors are found, contact Oracle Customer Support.
3509:ERROR	Convert buffer type(<text>,<text>) failed, length too big! DESCRIPTION	The conversion of the buffer received from the remote host was unsuccessful due to the resulting size exceeding the maximum of 32767.
	ACTION	Examine the application to determine if it has an error, examine the DMCONFIG definition for any errors in the buffer type specification. If no errors are found, contact Oracle Customer Support.
3510:ERROR	Failed to tmalloc() buffer (<text>,<text>) of size <number> DESCRIPTION	No buffer was returned by tmalloc for the size specified to hold the converted buffer data.
	ACTION	Examine the application to determine if it has an error, otherwise contact Oracle Customer Support.
3511:ERROR	Missing subtype for FML type buffer! DESCRIPTION	The buffer type specified of FML requires that a subtype be specified.
	ACTION	Examine the application and the configuration to determine which buffer type is being used. Oracle Tuxedo Mainframe Adapter for SNA Gateway only supports standard Tuxedo buffer types over the domain gateway. If the buffer type is VIEW, VIEW32, then also examine the environment variables VIEWDIR, VIEWFILES, VIEW32DIR, or VIEW32FILES.
3512:ERROR	Malloc temporary buffer of size(<number>) failed! DESCRIPTION	An internal error has been detected in gw_nw_decode.
	ACTION	Contact Oracle Customer Support.
3513:ERROR	Malloc failed, unable to convert!	

	DESCRIPTION	The conversion of the buffer received from the remote host was unsuccessful.
	ACTION	Examine the application to determine if it has an error, examine the DMCONFIG definition for any errors in the buffer type specification. If no errors are found, contact Oracle Customer Support.
3514:ERROR	Failed to malloc temporary buffer of size <number>	
	DESCRIPTION	An internal error has been detected in gw_nw_decode.
	ACTION	Contact Oracle Customer Support.
3515:ERROR	Convert buffer type(<text>,<text>) failed, convert failure!	
	DESCRIPTION	The conversion of the buffer received from the remote host was unsuccessful.
	ACTION	Examine the application to determine if it has an error, examine the DMCONFIG definition for any mistakes in the buffer type specification. If no errors are found, contact Oracle Customer Support.
3516:ERROR	Failed to tmalloc() buffer (<text>,<text>) of size <number>	
	DESCRIPTION	No buffer was returned by tmalloc for the size specified to hold the converted buffer data.
	ACTION	Check the application to determine if it has an error, otherwise contact Oracle Customer Support.
3517:ERROR	Missing subtype definition for FML32 buffer type!	
	DESCRIPTION	The buffer type specified of FML32 requires that a subtype be specified.
	ACTION	Correct the DMCONFIG definition.
3518:ERROR	Malloc temporary of size(<number>) failed!	
	DESCRIPTION	An internal error has been detected in gw_nw_decode.
	ACTION	Contact Oracle Customer Support.
3519:ERROR	Convert FML32 buffer type(<text>,<text>) failed, convert failure!	
	DESCRIPTION	The conversion of the buffer received from the remote host was unsuccessful.
	ACTION	Examine the application to determine if it has an error, examine the DMCONFIG definition for any errors in the buffer type specification. If no errors are found, contact Oracle Customer Support.
3520:ERROR	Malloc temporary buffer of size(<number>) failed!	
	DESCRIPTION	An internal error has been detected in gw_nw_decode.
	ACTION	Contact Oracle Customer Support.
3521:ERROR	Convert FML32 buffer with subtype(<text>) failed!	
	DESCRIPTION	The conversion of the buffer received from the remote host was unsuccessful.
	ACTION	Examine the application to determine if it has an error, examine the DMCONFIG definition for any errors in the buffer type specification. If no errors are found, contact Oracle Customer Support.
3522:ERROR	Failed to tmalloc() FML32 buffer of size <number> for <text>!	
	DESCRIPTION	An internal error has been detected in gw_nw_decode.
	ACTION	Contact Oracle Customer Support.
3523:ERROR	Fail to convert for FML buffer(<text>), exceeding limit!	
	DESCRIPTION	Failed to convert the FML buffer; the buffer size required to process the conversion exceeds the limit.

	ACTION	Examine the application and the view for this buffer for any errors. If no errors are found, contact Oracle Customer Support.
3524:ERROR	Fail to convert for FML32 buffer(<text>), exceeding limit!	
	DESCRIPTION	Failed to convert the FML32 buffer; the buffer size required to process the conversion exceeds the limit.
	ACTION	Examine the application and the view for this buffer for any errors. If no errors are found, contact Oracle Customer Support.
3525:ERROR	Fail to retrieve remote service definition!	
	DESCRIPTION	Unable to find the remote service definition in shared memory.
	ACTION	Examine the DMCONFIG remote service definitions. Correct any errors found. If no error is found, contact Oracle Customer Support.
3526:ERROR	Fail to retrieve local service definition!	
	DESCRIPTION	Unable to find the local service definition in shared memory.
	ACTION	Examine the DMCONFIG local service definitions. Correct any errors found. If no error is found, contact Oracle Customer Support.
3527:ERROR	Must specify subtype for <text>	
	DESCRIPTION	The buffer type specified of VIEW, VIEW32, X_C_TYPE, or X_COMMON requires that a subtype be specified.
	ACTION	Correct any errors in the DMCONFIG buffer type definitions. If no errors are found, contact Oracle Customer Support.
3528:ERROR	Malloc temporary buffer of size(<number>) failed!	
	DESCRIPTION	An internal error has been detected in gw_nw_encode.
	ACTION	Contact Oracle Customer Support.
3529:ERROR	Convert buffer type <text> with subtype <text> failed(rc=<number>)! DESCRIPTION	The conversion of the buffer received from Tuxedo was unsuccessful.
	ACTION	Examine the application to determine if it has an error, examine the DMCONFIG definition for any errors in the buffer type specification. If no errors are found, contact Oracle Customer Support.
3530:ERROR	Convert buffer type <text> with subtype <text> failed(rc=<number>)! DESCRIPTION	The conversion of the buffer received from Tuxedo was unsuccessful due to the resulting size exceeding the maximum of 32767.
	ACTION	Examine the application to determine if it has an error, examine the DMCONFIG definition for any errors in the buffer type specification. If no errors are found, contact Oracle Customer Support.
3531:ERROR	Malloc temporary buffer of size(<number>) failed!	
	DESCRIPTION	An internal error has been detected in gw_nw_encode.
	ACTION	Contact Oracle Customer Support.
3532:ERROR	FML buffer type requires subtype to be specified!	
	DESCRIPTION	The buffer type specified of FML requires that a subtype be specified.
	ACTION	Correct the DMCONFIG definition.
3533:ERROR	Malloc temporary buffer of size(<number>) failed!	
	DESCRIPTION	An internal error has been detected in gw_nw_encode.

	ACTION	Contact Oracle Customer Support.
3534:ERROR	Convert FML buffer with subtype <text> failed(Error=<number>)! DESCRIPTION	The conversion of the FML buffer received from Tuxedo was unsuccessful. Error is the return code from the FML conversion routine.
	ACTION	Examine the application to determine if it has an error, examine the DMCONFIG definition for any errors in the buffer type specification. If no errors are found, contact Oracle Customer Support.
3535:ERROR	Malloc temporary buffer of size(<number>) failed! DESCRIPTION	An internal error has been detected in <code>gw_nw_encode</code> .
	ACTION	Contact Oracle Customer Support.
3536:ERROR	Convert FML buffer with subtype <text> failed(Error=<number>)! DESCRIPTION	The conversion of the FML buffer received from Tuxedo was unsuccessful. Error is the return code from the FML conversion routine.
	ACTION	Examine the application to determine if it has an error, examine the DMCONFIG definition for any errors in the buffer type specification. If no errors are found, contact Oracle Customer Support.
3537:ERROR	Malloc buffer of size(<number>) failed! DESCRIPTION	An internal error has been detected in <code>gw_nw_encode</code> .
	ACTION	Contact Oracle Customer Support.
3538:ERROR	FML32 buffer type requires subtype to be specified! DESCRIPTION	The buffer type specified of FML32 requires that a subtype be specified.
	ACTION	Correct the DMCONFIG definition.
3539:ERROR	Malloc temporary buffer of size(<number>) failed! DESCRIPTION	An internal error has been detected, <code>gw_nw_Alloc</code> was unable to allocate memory needed.
	ACTION	Contact Oracle Customer Support.
3540:ERROR	Convert FML32 buffer with subtype <text> failed(Error=<number>)! DESCRIPTION	The conversion of the FML32 buffer received from Tuxedo was unsuccessful. Error is the return code from the FML conversion routine.
	ACTION	Examine the application to determine if it has an error, examine the DMCONFIG definition for any errors in the buffer type specification. If no errors are found, contact Oracle Customer Support.
3541:ERROR	Malloc temporary buffer of size(<number>) failed! DESCRIPTION	An internal error has been detected, <code>gw_nw_Alloc</code> was unable to allocate memory needed.
	ACTION	Contact Oracle Customer Support.
3542:ERROR	Convert FML32 buffer with subtype <text> failed(Error=<number>)! DESCRIPTION	The conversion of the FML32 buffer received from Tuxedo was unsuccessful. Error is the return code from the FML conversion routine.
	ACTION	Examine the application to determine if it has an error, examine the DMCONFIG definition for any errors in the buffer type specification. If no errors are found, contact Oracle Customer Support.
4000:ERROR	GP_SEND flow controlled for GPND <number>	

	DESCRIPTION	An internal error has been detected, <code>gw_nw_proto_flow</code> has detected a flow control callback from <code>gp_send</code> .
	ACTION	Contact Oracle Customer Support.
4100:ERROR	Maximum number of pointers exceeded	
	DESCRIPTION	An internal error has been detected. The maximum number of pointers has exceeded 500.
	ACTION	Contact Oracle Customer Support.
4124:ERROR	Unable to format CRM_acall	
	DESCRIPTION	An internal error has been detected, <code>CRM_acall</code> has returned an error.
	ACTION	Contact Oracle Customer Support.
4125:ERROR	Unable to send data on sna_conv_idx <number>, gpnd = <number>	
	DESCRIPTION	An error has occurred while sending acall data (<code>gw_nw_acall</code>) to the CRM.
	ACTION	Examine the ULOG for more information concerning the reason for the failure.
4126:ERROR	Unable to format CRM_connect	
	DESCRIPTION	An internal error has been detected, <code>CRM_connect</code> has returned an error.
	ACTION	Contact Oracle Customer Support.
4127:ERROR	Unable to send data on sna_conv_idx <number>, gpnd = <number>	
	DESCRIPTION	An error has occurred while sending connect data (<code>gw_nw_connect</code>) to the CRM.
	ACTION	Examine the ULOG for more information concerning the reason for the failure.
4128:ERROR	Unable to format CRM_send_data	
	DESCRIPTION	An internal error has been detected, <code>CRM_send</code> has returned an error.
	ACTION	Contact Oracle Customer Support.
4130:ERROR		
	DESCRIPTION	An internal error has been detected, <code>CRM_connect_rsp</code> has returned an error.
	ACTION	Contact Oracle Customer Support
5019:ERROR	Send event for RRCORR() failed!	
	DESCRIPTION	An internal error has been detected, <code>gw_nw_SendEvent</code> has returned an error.
	ACTION	Contact Oracle Customer Support.
5020:ERROR	Encode ACALL_RSP failed, RRCORR(<number>)! 	
	DESCRIPTION	An internal error has been detected, <code>CRM_acall_rsp</code> has returned an error.
	ACTION	Contact Oracle Customer Support.
5021:ERROR	Encode reply failed!	
	DESCRIPTION	An error was detected while attempting to encode the reply message data destined for the remote host.
	ACTION	Examine the ULOG for more information concerning the reason for the failure.
5022:ERROR	Invalid network descriptor(<number>! 	
	DESCRIPTION	The network descriptor this context is invalid.
	ACTION	Examine the ULOG for more information concerning the reason for the failure.

5023:ERROR	Could not get transaction
	DESCRIPTION Failed to associate a transactional ACALL service request to a transaction object. The most likely reasons for this failure are: <ul style="list-style-type: none"> • The locally configured limit has been exceeded, or • Memory allocation has failed to create the transaction object.
	ACTION Determine if the application is licensed to do transactions. Examine the DMCONFIG file for the MAXTRAN parameter, this number is the limit. If MAXTRAN is not specified, then examine the ubbconfig file for the MAXGTT parameter. In the later case MAXGTT is the limit. Modify the limit if necessary. Verify that the remote domain MAXSYNCLVL is set to 2.
5024:ERROR	Could not get API
	DESCRIPTION An internal error has been detected. The API information is invalid.
	ACTION Contact Oracle Customer Support.
5025:ERROR	Could not get transaction
	DESCRIPTION Failed to associate a transactional CONNECT service request to a transaction object. The most likely reasons for this failure are: <ul style="list-style-type: none"> • The locally configured limit has been exceeded, or • Memory allocation has failed to create the transaction object
	ACTION Determine if the application is licensed to do transactions. Examine the DMCONFIG file for the MAXTRAN parameter, this number is the limit. If MAXTRAN is not specified, then examine the ubbconfig file for the MAXGTT parameter. In the later case MAXGTT is the limit. Modify the limit if necessary. Verify that the remote domain MAXSYNCLVL is set to 2.
5026:ERROR	Could not get API
	DESCRIPTION An internal error has been detected. The API information is invalid.
	ACTION Contact Oracle Customer Support.
5027:ERROR	Could not get API
	DESCRIPTION An internal error has been detected. The API information is invalid.
	ACTION Contact Oracle Customer Support.
5028:ERROR	Protocol violation, shutdown Gateway!
	DESCRIPTION An internal error has been detected by gw_nw_crmrcv.
	ACTION Contact Oracle Customer Support.
5029:ERROR	Link status down for remote domain <text>
	DESCRIPTION The link status returned by the CRM is LINK_DOWN.
	ACTION Determine if the stack is active and check that the session between the stack provider and the remote host is active.
5030:ERROR	Link status pending for remote domain <text>
	DESCRIPTION The link status returned by the CRM is LINK_PENDING. Transaction recovery for the link is still in progress following a restart. Only non-transactional requests may be sent to this link.
	ACTION Wait until recovery is complete before requesting transactional services on this link.
5031:ERROR	Transaction not allowed, request rejected!

	DESCRIPTION	Failed to send a transactional ACALL request to the remote domain because MAXSYNCLVL is not 2.
	ACTION	Determine the following things: <ul style="list-style-type: none"> • Oracle Tuxedo Mainframe Adapter for SNA Gateway is licensed for transactions. and • DMCONFIG specifies MAXSYNCLVL=2 for the link.
5032:ERROR	Transaction not allowed, request rejected!	
	DESCRIPTION	Failed to send a transactional CONNECT request to the remote domain because MAXSYNCLVL is not 2.
	ACTION	Determine the following things: <ul style="list-style-type: none"> • Oracle Tuxedo Mainframe Adapter for SNA Gateway is licensed for transactions and • DMCONFIG specifies MAXSYNCLVL=2 for the link.
5033:ERROR	Protocol violation, shutdown Gateway!	
	DESCRIPTION	An internal error has been detected by gw_nw_crmrcv.
	ACTION	Contact Oracle Customer Support.
5034:ERROR	Allocate context failed!	
	DESCRIPTION	An internal error has been detected. An error was returned by gw_nw_AllocNwCtx.
	ACTION	Contact Oracle Customer Support.
5035:ERROR	Enter shutdown	
	DESCRIPTION	An internal error has been detected by gw_nw_crmrcv.
	ACTION	Contact Oracle Customer Support.
5036:ERROR	Invalid conversation identification, shutdown Gateway!	
	DESCRIPTION	An internal error has been detected by gw_nw_convsend.
	ACTION	Contact Oracle Customer Support.
5037:ERROR	Unknown message type, shutdown Gateway!	
	DESCRIPTION	An internal error has been detected by gw_nw_crmrcv.
	ACTION	Contact Oracle Customer Support.
5038:ERROR	LINK_STATUS state error, shutdown Gateway!	
	DESCRIPTION	An internal error has been detected by gw_nw_crmrcv.
	ACTION	Contact Oracle Customer Support.
5039:ERROR	Protocol violation, shutdown Gateway!	
	DESCRIPTION	An internal error has been detected by gw_nw_crmrcv.
	ACTION	Contact Oracle Customer Support.
5040:ERROR	Enter shutdown	
	DESCRIPTION	An internal error has been detected, an error was returned by gw_nw_getsncrm_idx.
	ACTION	Contact Oracle Customer Support.
5041:ERROR	Trans Event Error shutdown	
	DESCRIPTION	An internal error has been detected, an error was returned by gw_nw_TransetUpExtTmsEvent.
	ACTION	Contact Oracle Customer Support.
5042:WARNING	SIGNON identify failed, shutdown Gateway!	
	DESCRIPTION	CRM returned an error to the SIGNON request by Oracle Tuxedo Mainframe Adapter for SNA Gateway.
	ACTION	Examine the ULOG for more information concerning the reason for the failure.
5044:ERROR	SIGNON state error, shutdown Gateway!	

	DESCRIPTION	An internal error has been detected, an error was returned by <code>_gw_nw_CRM_nextop</code> .
	ACTION	Contact Oracle Customer Support.
5045:ERROR	STARTUP state error, shutdown Gateway!	
	DESCRIPTION	An internal error has been detected, an error was returned by <code>_gw_nw_CRM_nextop</code> .
	ACTION	Contact Oracle Customer Support.
5046:WARNING	LINK_STATUS decode error, shutdown Gateway!	
	DESCRIPTION	An internal error has been detected. An error was returned by <code>CRM_linkstatus_rcv</code> .
	ACTION	Contact Oracle Customer Support.
5047:WARNING	LINK_STATUS status error, shutdown Gateway!	
	DESCRIPTION	An internal error has been detected by <code>gw_nw_crmrcv</code> .
	ACTION	Contact Oracle Customer Support.
5048:WARNING	LINK_CONFIG state error, shutdown Gateway!	
	DESCRIPTION	An internal error has been detected, an error was returned by <code>_gw_nw_CRM_nextop</code> .
	ACTION	Contact Oracle Customer Support.
5049:WARNING	STACK_CONFIG state error, shutdown Gateway!	
	DESCRIPTION	An internal error has been detected, an error was returned by <code>_gw_nw_CRM_nextop</code> .
	ACTION	Contact Oracle Customer Support.
5050:WARNING	SHUTDOWN message received, shutdown Gateway!	
	DESCRIPTION	<code>GWSNAX</code> has received a SHUTDOWN request from the CRM and will shut down immediately.
	ACTION	Examine the ULOG for more information concerning the reason for the shutdown.
5051:ERROR	LOG_DATA decode error, shutdown Gateway!	
	DESCRIPTION	An internal error has been detected. An error was returned by <code>CRM_log_data_rcv</code> .
	ACTION	Contact Oracle Customer Support.
5052:ERROR	LOG_DATA state error(<number>), shutdown Gateway!	
	DESCRIPTION	An internal error has been detected by <code>gw_nw_crmrcv</code> .
	ACTION	Contact Oracle Customer Support.
5053:ERROR	Protocol violation, shutdown Gateway!	
	DESCRIPTION	An internal error has been detected by <code>gw_nw_crmrcv</code> . Inbound ACALL request contained <code>SEND_INVITE</code> for function DPL.
	ACTION	Contact Oracle Customer Support.
5054:ERROR	Invalid action index	
	DESCRIPTION	An internal error has been detected by <code>gw_nw_InBoundConnectRqst</code> .
	ACTION	Contact Oracle Customer Support.
5055:ERROR	Invalid action index	
	DESCRIPTION	An internal error has been detected by <code>gw_nw_OutBoundConnectRsp</code> .
	ACTION	Contact Oracle Customer Support.
5056:ERROR	Protocol violation, shutdown Gateway!	
	DESCRIPTION	An internal error has been detected by <code>gw_nw_crmrcv</code> .

	ACTION	Contact Oracle Customer Support.
5057:ERROR	SEND_DATA decode error, shutdown Gateway!	
	DESCRIPTION	An internal error has been detected by gw_nw_crmrcv.
	ACTION	Contact Oracle Customer Support.
5058:ERROR	Cannot create action to send remote disconnect	
	DESCRIPTION	An internal error has been detected, an error was returned by gw_new_action.
	ACTION	Contact Oracle Customer Support.
5059:ERROR	ACALL decode error, shutdown Gateway!	
	DESCRIPTION	An internal error has been detected by gw_nw_crmrcv.
	ACTION	Contact Oracle Customer Support.
5060:ERROR	Invalid action index	
	DESCRIPTION	An internal error has been detected by gw_nw_InBoundAcallRqst.
	ACTION	Contact Oracle Customer Support.
5061:ERROR	Invalid action index	
	DESCRIPTION	An internal error has been detected by gw_nw_OutBoundAcallRsp.
	ACTION	Contact Oracle Customer Support.
5062:ERROR	Failed to find local service(<text>) definition!	
	DESCRIPTION	Unable to find the local service definition for the inbound request.
	ACTION	Examine the DMCONFIG local service definitions. Correct any errors found. If no error is found, contact Oracle Customer Support.
5063:ERROR	Invalid link index	
	DESCRIPTION	An internal error has been detected, an error was returned by gw_nw_set_link_idx.
	ACTION	Contact Oracle Customer Support.
5064:ERROR	Invalid query_rsp index	
	DESCRIPTION	An internal error has been detected by gw_nw_QueryProcess.
	ACTION	Contact Oracle Customer Support.
5065:ERROR	Enter shutdown	
	DESCRIPTION	A failure has been detected while sending the CRM_QSVC_RSP verb to the CRM.
	ACTION	Examine the ULOG for more information concerning the reason for the failure. Determine if the CRM process is still active.
5066:ERROR	Encode SEND_DATA failed, RRCORR(<number>)! 	
	DESCRIPTION	An internal error has been detected by gw_nw_reply.
	ACTION	Contact Oracle Customer Support.
5067:ERROR	Decode inbound ACALL request failed!	
	DESCRIPTION	An error has occurred while decoding an inbound ACALL request in gw_nw_decode.
	ACTION	Examine the ULOG for more information concerning the reason for the failure.
5068:ERROR	Could not get FUNC	
	DESCRIPTION	An internal error has been detected by gw_nw_acall.

	ACTION	Contact Oracle Customer Support.
5069:ERROR	Could not get FUNC	
	DESCRIPTION	An internal error has been detected by <code>gw_nw_connect</code> .
	ACTION	Contact Oracle Customer Support.
5070:ERROR	SECURITY: Failed in security checking for RRCORR(<number>)! 	
	DESCRIPTION	Security checking failed for the inbound ACALL request. An error was returned by <code>gw_nw_SetSecurity</code> .
	ACTION	Examine the ULOG for more information concerning the reason for the failure.
5071:ERROR	BAD Transaction ID shutdown	
	DESCRIPTION	An internal error has been detected by <code>gw_nw_SendData</code> .
	ACTION	Contact Oracle Customer Support.
5072:ERROR	BAD Transaction ID shutdown	
	DESCRIPTION	An internal error has been detected by <code>gw_nw_OutBoundAcallRsp</code> .
	ACTION	Contact Oracle Customer Support.
5073:ERROR	Could not get API	
	DESCRIPTION	An internal error has been detected by <code>gw_nw_QueryProcess</code> .
	ACTION	Contact Oracle Customer Support.
5074:ERROR	Decode outbound ACALL response failed, RRCORR(<number>)! 	
	DESCRIPTION	An error has occurred while decoding the response from the host application to an outbound ACALL.
	ACTION	Examine the ULOG for more information concerning the reason for the failure.
5075:ERROR	Invalid query_rsp index	
	DESCRIPTION	An internal error has been detected by <code>gw_nw_ConnectError</code> .
	ACTION	Contact Oracle Customer Support.
5077:ERROR	Enter shutdown	
	DESCRIPTION	A failure has been detected while sending the CRM_QSVC_RSP verb to the CRM.
	ACTION	Examine the ULOG for more information concerning the reason for the failure. Determine if the CRM process is still active.
5078:ERROR	Enter shutdown	
	DESCRIPTION	A failure has been detected while sending the CRM_CONNECT_RSP verb to the CRM.
	ACTION	Examine the ULOG for more information concerning the reason for the failure. Determine if the CRM process is still active.
5079:ERROR	BAD Transaction ID shutdown	
	DESCRIPTION	An internal error has been detected by <code>gw_nw_InBoundAcallRqst</code> .
	ACTION	Contact Oracle Customer Support.
5080:ERROR	Decode inbound connect request failed, RRCORR(<number>)! 	
	DESCRIPTION	An error has occurred while decoding an inbound connect request in <code>gw_nw_InBoundConnectRqst</code> .
	ACTION	Examine the ULOG for more information concerning the reason for the failure.

5081:ERROR	<p>SECURITY: Inbound connect request security check failed, RRCORR(<number>)! DESCRIPTION Security checking failed for the inbound CONNECT request. An error was returned by <code>gw_nw_SetSecurity</code>.</p> <p>ACTION Examine the ULOG for more information concerning the reason for the failure.</p>
5082:ERROR	<p>Decode SEND_DATA failed, RRCORR(<number>)! DESCRIPTION An error has occurred while decoding an outbound send data request in <code>gw_nw_SendData</code>.</p> <p>ACTION Examine the ULOG for more information concerning the reason for the failure.</p>
5083:ERROR	<p>Invalid buffer type, encode failed! DESCRIPTION The buffer type specified for the outbound message was invalid. Oracle Tuxedo Mainframe Adapter for SNA Gateway only supports standard Tuxedo buffer types over the domain gateway.</p> <p>ACTION Correct the DMCONFIG definition.</p>
5084:ERROR	<p>Outbound ACALL encode failed! DESCRIPTION An error has occurred while encoding an outbound ACALL request in <code>gw_nw_encode</code>.</p> <p>ACTION Examine the ULOG for more information concerning the reason for the failure.</p>
5085:ERROR	<p>BAD CRM_IDX shutdown DESCRIPTION An internal error has been detected by <code>gw_nw_InBoundAcallRqst</code>.</p> <p>ACTION Contact Oracle Customer Support.</p>
5086:ERROR	<p>BAD Context shutdown DESCRIPTION An internal error has been detected by <code>gw_nw_InBoundAcallRqst</code>.</p> <p>ACTION Contact Oracle Customer Support.</p>
5087:ERROR	<p>Invalid buffer type, decode failed! DESCRIPTION The buffer type specified for the inbound message was invalid. Oracle Tuxedo Mainframe Adapter for SNA Gateway only supports standard Tuxedo buffer types over the domain gateway.</p> <p>ACTION Correct the DMCONFIG definition.</p>
5088:ERROR	<p>Unable to get RDOM DESCRIPTION An internal error has been detected by <code>gw_nw_init_env</code>.</p> <p>ACTION Contact Oracle Customer Support.</p>
5089:ERROR	<p>BAD CRM_IDX shutdown DESCRIPTION An internal error has been detected by <code>gw_nw_InBoundConnectRqst</code>.</p> <p>ACTION Contact Oracle Customer Support.</p>
5090:ERROR	<p>Outbound CONNECT encode failed! DESCRIPTION An error has occurred while encoding an outbound CONNECT request in <code>gw_nw_encode</code>.</p> <p>ACTION Examine the ULOG for more information concerning the reason for the failure.</p>
5091:ERROR	<p>Outbound CONVSEND encode failed! DESCRIPTION An error has occurred while encoding an outbound CONVSEND request in <code>gw_nw_encode</code>.</p>

	ACTION	Examine the ULOG for more information concerning the reason for the failure.
5093:ERROR	Bad Connect	
	DESCRIPTION	An internal error has been detected by <code>gw_nw_crmrcv</code> .
	ACTION	Contact Oracle Customer Support.
5094:ERROR	Unable to send data on sna_conv_idx <number>, gpnd = <number>	
	DESCRIPTION	An error has occurred while sending a connect response to the CRM.
	ACTION	Examine the ULOG for more information concerning the reason for the failure.
5095:ERROR	BAD Transaction ID shutdown	
	DESCRIPTION	An internal error has been detected by <code>gw_nw_InBoundConnectRqst</code> .
	ACTION	Contact Oracle Customer Support.
5096:ERROR	Unable to obtain socket id	
	DESCRIPTION	An internal error has been detected by <code>gw_nw_link_gp</code> .
	ACTION	Contact Oracle Customer Support.
5097:ERROR	Unable to get RDOM	
	DESCRIPTION	An internal error has been detected by <code>gw_nw_link_gp</code> .
	ACTION	Contact Oracle Customer Support.
5098:ERROR	Invalid action index	
	DESCRIPTION	An internal error has been detected by <code>gw_nw_link_gp</code> .
	ACTION	Contact Oracle Customer Support.
5099:ERROR	Allocate context failed!	
	DESCRIPTION	An internal error has been detected, an error was returned by <code>gw_nw_AllocNwCtx</code> .
	ACTION	Contact Oracle Customer Support.
5100:ERROR	Unable to obtain socket id	
	DESCRIPTION	An internal error has been detected by <code>gw_nw_init_env</code> .
	ACTION	Contact Oracle Customer Support.
5101:ERROR	Unable to find view file for <text> <text>	
	DESCRIPTION	Failed to find the view file for the FML buffer type specified in the INBUFTYPE for an inbound request.
	ACTION	Examine the environment variables VIEWDIR, VIEWFILES, VIEW32DIR, and VIEW32FILES to determine if the requested view is present.
5102:ERROR	Invalid action index	
	DESCRIPTION	An internal error has been detected by <code>gw_nw_reply</code> .
	ACTION	Contact Oracle Customer Support.
5103:ERROR	Invalid action index	
	DESCRIPTION	An internal error has been detected by <code>gw_nw_reply</code> .
	ACTION	Contact Oracle Customer Support.
5104:ERROR	Invalid action index	
	DESCRIPTION	An internal error has been detected by <code>gw_nw_QueryProcess</code> .
	ACTION	Contact Oracle Customer Support.
5105:ERROR	Invalid buffer pointer	
	DESCRIPTION	An internal error has been detected by <code>gw_nw_encode</code> .
	ACTION	Contact Oracle Customer Support.

5106:ERROR	Allocate send buffer failed!
	DESCRIPTION An internal error has been detected, an error was returned by <code>gw_nw_empty_sndbuf</code> .
	ACTION Contact Oracle Customer Support.
5107:ERROR	Unable to get appkey for this outbound request
	DESCRIPTION An internal error has been detected, <code>gw_shm_getbylink</code> returned an error while getting the remote domain entry.
	ACTION Examine the DMCONFIG remote domain definitions and its security setting. Correct any errors found. If no error is found, contact Oracle Customer Support.
5108:ERROR	Enter shutdown
	DESCRIPTION An internal error has been detected by <code>gw_nw_SendEvent</code> .
	ACTION Contact Oracle Customer Support.
5109:ERROR	Unable to find view file for <text> <text>
	DESCRIPTION Failed to find the view file for encoding outbound FML data as specified in the OUTBUFTYPE.
	ACTION Examine the environment variables VIEWDIR, VIEWFILES, VIEW32DIR, and VIEW32FILES to determine if the requested view is present.
5110:ERROR	Unable to find view file for <text> <text>
	DESCRIPTION Failed to find the view file for encoding outbound FML32 data as specified in the OUTBUFTYPE.
	ACTION Examine the environment variables VIEWDIR, VIEWFILES, VIEW32DIR, and VIEW32FILES to determine if the requested view is present.
5111:ERROR	BAD Context shutdown caused at transaction associate
	DESCRIPTION An internal error has been detected, an error was returned by <code>gw_nw_AssociateTranByTCTXT</code> .
	ACTION Contact Oracle Customer Support.
5112:ERROR	System full, can't create action, Enter shutdown
	DESCRIPTION Failed to create GWSNAX Gateway internal scheduled event to handle failure ore disconnect.
	ACTION Either memory allocation failed or a system limit was reached. Contact Oracle Customer Support.
5113:ERROR	Realloc buffer of size(<number>) failed!
	DESCRIPTION An internal error has been detected, <code>gw_nw_Realloc</code> was unable to allocate memory needed.
	ACTION Contact Oracle Customer Support.
5114:ERROR	Invalid network descriptor(<number>)!
	DESCRIPTION An internal error has been detected by <code>gw_nw_acall</code> .
	ACTION Contact Oracle Customer Support.
5115:ERROR	Encryption setup failed, code <number>
	DESCRIPTION An error has occurred while establishing link-level encryption with the CRM.
	-1: The processes are using different encryption types (ex. GPE)
	6: The processes are using different versions of the encryption protocol.
	7: The negotiation of an encryption level has failed.
	8: A system error has occurred in the encryption setup

	ACTION	Verify that the setup of encryption on both sides of the link is correct. Verify that there are common encryption levels in the ranges specified on the process command lines. Verify that the correct encryption libraries are installed.
5116:ERROR	CRM authentication setup failed	
	DESCRIPTION	An error has occurred while authenticating a CRM client.
	ACTION	Verify that the authentication file(s) for both the CRM and the client are set up correctly, and that both have correct access privileges. Verify that the authentication file is specified correctly on both command lines.
5117:INFO	Link encryption established, <text> bits	
	DESCRIPTION	The SNAX has established link-level encryption with the CRM.
	ACTION	No action is required. If the encryption level specified does not match the desired value, verify the encryption level range specified on the SNAX and CRM command lines. A value of '40/56' indicates that encryption is set at 56, even though the specified range may be as low as 40.
6004:ERROR	Fails to propagate USERID for inbound request, RRCORR(%lx)!	
	DESCRIPTION	Fails to propagate USERID for inbound call.
	ACTION	Check the DMCONFIG and CICS security configuration to determine if it has an error, otherwise contact Oracle Customer Support.
6005:ERROR	Fails to propagate USERID for conversational inbound request, RRCORR(%lx)!	
	DESCRIPTION	Fails to propagate USERID for conversational inbound call.
	ACTION	Check the DMCONFIG and CICS security configuration to determine if it has an error, otherwise contact Oracle Customer Support.
6006: ERROR	TDQ encoding failed	
	DESCRIPTION	Fails to encode TDQ message.
	ACTION	Contact ORACLE Customer Support.
6007:ERROR	TDQ decoding failed	
	DESCRIPTION	Fails to decode TDQ message.
	ACTION	Contact ORACLE Customer Support.
6008:ERROR	Fails to execute TDQ command	
	DESCRIPTION	Fails to execute TDQ command.
	ACTION	Check ULOG for additional information.
6011:WARN	Input buffer is bigger than RECORD, received(%ld), configured RECORD length(%ld)	
	DESCRIPTION	The RECORD buffer received of the size specified is smaller than the actual RECORD length configured.
	ACTION	Check the DMCONFIG to determine if it has an error, otherwise contact Oracle Customer Support.
6012:ERROR	Both INBUFTYPE and OUTBUFTYPE should be RECORD	
	DESCRIPTION	The Gateway can't handle the case if INBUFTYPE and OUTBUFTYPE are not all RECORD.
	ACTION	Change both INBUFTYPE and OUTBUFTYPE to RECORD.
9001:ERROR		<taskname> timed out with failCode <failcode>

	DESCRIPTION	A conversation has timed out in the CRM with the stack return code of <code><failcode></code> . A timer event set to watch a conversation has expired. <code><taskname></code> may appear as: OB-Conversation <code>#nn</code> (<linkref>) tx <code>#m</code> <code><tranname></code> , or IB-Conversation <code>#nn</code> (<linkref>) tx <code>#m</code> <code><tranname></code> where: <code>nn</code> is an internal APPC conversation number and <code>m</code> is the transaction context where -1 signifies non-transactional. Outbound Conversation <code>nnnn</code> Link (<linkref>) TCTXT (<code>tctxt</code>) Tran (<code><tranname></code>), or Inbound Conversation <code>nnnn</code> Link (<linkref>) TCTXT (<code>tctxt</code>) Tran(<code><tranname></code>) where <code>nnnn</code> is the CRM Logical Task Number and <code>tctxt</code> is the transaction context, where FFFFFFFF indicates notn-transactional.
	ACTION	Examine <code>stderr</code> and the ULOG for additional information concerning the failure.
9002:ERROR	Unable to create APPC Server for <code><text></code>.	
	DESCRIPTION	CRM was unable to instantiate the stack object due to an error.
	ACTION	Check for additional messages in <code>stderr</code> . The shared library for the stack or the stack interface might not have been loaded due to an incorrect library path.
9003:ERROR	Server Failed (<stackref>), Code = <code><returncode></code>	
	DESCRIPTION	CRM received a bad return code from the stack start-up.
	ACTION	The <code><returncode></code> is the value returned by the SNA Stack software. Check the status of the stack, the configuration of the stack, and the gateway configuration. Note: Unless you started the CRM as a Tuxedo server, you must manually kill the CRM process.
9004:ERROR	Configuration change on link <linkref> requires cold start	
	DESCRIPTION	Attempting to do a warm start after changing the domain configuration.
	ACTION	Change start type to COLD and restart.
9005:WARNING	Unrecovered transaction, TCTXT(<code><text></code>), blob dropped. Transaction presumed forgotten.	
	DESCRIPTION	An attempt was made by Tuxedo to recover the specified transaction that was unknown by the CRM. It is presumed that it was already committed or aborted prior to the recovery attempt.
	ACTION	None. This message is for information only.
9006:ERROR	Unable to start the recovery task for link <linkref>	
	DESCRIPTION	An error occurred during the warm start of Tuxedo.
	ACTION	Cold start the Tuxedo application.

9008:WARNING	Unknown tranid dropped, id=<tctxt>
	DESCRIPTION Recovery was requested by Tuxedo on a transaction that was already forgotten by the CRM.
	ACTION None. This message is for information only.
9009:ERROR	No blob with recovery request. Transaction dropped, id=<tctxt>
	DESCRIPTION Attempting to warm start after the CRM's BLOBLOG has been modified.
	ACTION Change start type to COLD and restart.
9010:ERROR	<taskname> failed with failCode <failcode>
	DESCRIPTION A conversation has failed with the stack return code of <failcode>.
	<taskname> may appear as:
	Outbound Conversation <i>nnnn</i> Link <linkref>
	TCTXT (<tctxt>) Tran <traname>, or
	Inbound Conversation <i>nnnn</i> Link <linkref>
	TCTXT (<tctxt>) Tran <traname>,
	where:
	<i>nnnn</i> is the CRM Logical Task Number.
	<i>tctxt</i> is the transaction context where
	<i>FFFFFFFF</i> signifies non-transactional.
	Possible values for the <failcode>are:
	<ol style="list-style-type: none"> 1. Communications - unable to create the APPCserver object. 2. MemoryAllocation - internal error allocating memory. 3. InvalidObject - a CRM object could not be created or has been made invalid by some previous error. 4. InputOutput - error occurred during file I/O or an unexpected APPC return code was received. 5. Registration - internal task cannot be registered
	ACTION Examine <code>stderr</code> and the <code>ULOG</code> for additional information concerning the failure. For failcode Input/Output, verify that the user starting the CRM process has the proper file permissions for the BLOBLOG and RSTRTLOG. If no apparent error is found, contact Oracle Customer Support.
9011:ERROR	Attempt to connect as second master refused!
	DESCRIPTION A second GWSNAX is attempting to connect to the CRM as a master gateway. Only one master gateway is allowed.
	ACTION Ensure that multiple Tuxedo configurations do not use the same CRM address.
9012:ERROR	Attempt to connect as master in autonomous mode refused!
	DESCRIPTION An attempt to connect to the CRM as a master gateway was made when the CRM was running in autonomous mode.
	ACTION Ensure that multiple Tuxedo configurations do not use the same CRM address.
9013:ERROR	Attempt to connect with incorrect group name (<groupname>) refused!
	DESCRIPTION The group name in the gateway configuration file does not match the group name specified in the CRM command line.

	ACTION	Correct the group name that is in error and restart.
9014:ERROR	INTERNAL ERROR: memory allocation failed [for new context/data buffer]	
	DESCRIPTION	INTERNAL ERROR: memory allocation failed [for new context/data buffer]
	ACTION	Contact Oracle Customer Support.
9015:ERROR	INTERNAL ERROR: server registration failed	
	DESCRIPTION	Internal error registering the APPC server. APPC libraries not found. The stack failed.
	ACTION	Contact Oracle Customer Support.
9016:ERROR	Link refers to undefined APPC stack (<stackref>)!	
	DESCRIPTION	The <code>stackref</code> in the link configuration is incorrect.
	ACTION	Correct the <code>stackref</code> that is in error, run <code>dmloadcf</code> , and restart.
9017:ERROR	INTERNAL ERROR: link registration failed	
	DESCRIPTION	Internal error registering the link. The stack failed.
	ACTION	Contact Oracle Customer Support.
9018:ERROR	Invalid Transition <additional information>	
	DESCRIPTION	An internal state table failure has occurred. The <additional information> will be one of the following:
		<ol style="list-style-type: none"> 1. From <oldstate> to <newstate> for <dir>-bound transaction TCTXT <tid> 2. To <newstate> for inbound transaction TCTXT <tid> 3. To <newstate> for outbound transaction TCTXT <tid>
	ACTION	Contact Oracle Customer Support.
9019:ERROR	Unknown Service Correlator = <correlator>, message dropped	
	DESCRIPTION	Internal error assigning service correlator values. Message context lost.
	ACTION	Contact Oracle Customer Support.
9020:ERROR	Duplicate Service Correlator = <correlator>	
	DESCRIPTION	Internal error assigning service correlator values.
	ACTION	Contact Oracle Customer Support.
9021:ERROR	Invalid Remote Link Name <linkref>	
	DESCRIPTION	The remote link name in a request does not match any defined link name.
	ACTION	Correct the gateway configuration and restart.
9022:ERROR	Invalid transaction context = <tctx>	
	DESCRIPTION	Internal error assigning transaction context values. Bad transaction.
	ACTION	Contact Oracle Customer Support.
9023:ERROR	Unknown Service Correlator = <correlator>, message dropped	
	DESCRIPTION	Internal error assigning service correlator values. Message context lost.
	ACTION	Contact Oracle Customer Support.
9024:ERROR	Invalid initial syncpoint received from subordinate, TCTXT(<text>)	
	DESCRIPTION	Syncpoint processing protocol violation. Subordinate member of conversation attempted to initiate a syncpoint. XA does not support syncpoints from subordinate members.

	ACTION	Contact Oracle Customer Support.
9025:ERROR	Invalid Input Message Discarded	
	DESCRIPTION	Internal error, bad message sent between GWSNAX and CRM. Possibly incompatible Oracle Tuxedo Mainframe Adapter for SNA Gateway and CRM.
	ACTION	Contact Oracle Customer Support.
9026:ERROR	CNOS Notification Received for unknown partner <partnerLU>	
	DESCRIPTION	Multiple instances of the CRM may be using the same local LU.
	ACTION	Ensure that multiple Tuxedo configurations do not use the same local LU.
9027:WARNING	Remote Stop Received for <linkref>	
	DESCRIPTION	The remote host has issued a stop for the specified link.
	ACTION	None. This message for information only.
9028:WARNING	Remote Start Received for <linkref>	
	DESCRIPTION	The remote host has issued a start for the specified link.
	ACTION	None. This message for information only.
9029:ERROR	Undefined Remote LU on link <linkref>	
	DESCRIPTION	The remote LU does not exist as defined.
	ACTION	Check the gateway configuration file and the stack configuration and correct the mis-match.
9030:ERROR	Unable to start session on link <linkref>. Reason=<reason>	
	DESCRIPTION	Link activation failure due to SNA error.
	ACTION	<reason> is the description of the stack return code. Determine the cause and correct.
9031:ERROR	Unable to initialize link <linkref>. Reason=<reason>	
	DESCRIPTION	Link initialization failure due to SNA error.
	ACTION	<reason> is the description of the stack return code. Determine the cause and correct.
9032:ERROR	No Available Session on link <linkref> for context <correlator>	
	DESCRIPTION	Max sessions has been exceeded.
	ACTION	Check session limits in gateway configuration, stack configuration, CICS or VTAM. Increase if necessary.
9033:ERROR	Requested Synclevel not supported by link <linkref> for context <correlator> (synclevel <level>)	
	DESCRIPTION	Attempted to issue a request at sync level <level> on a link that does not support that level.
	ACTION	Correct application or gateway configuration.
9034:ERROR	Service Request at SyncLevel=2 Rejected on PENDING link <linkref> for context <correlator>	
	DESCRIPTION	An attempt to start a new sync level 2 request has been received and the Link is currently processing recovery information.
	ACTION	Wait until recovery is complete to request sync level 2 services.
9035:ERROR	Inbound Request Transform Failed (<status>) for context <correlator>	

	DESCRIPTION	An error has occurred while processing the CICS transform for an inbound DPL request. This normally occurs when the API entry in the gateway configuration for the local service specifies CICS instead of ATMI.
	ACTION	Check gateway configuration for incorrect specification of local service API entry.
9036:ERROR	Inbound Response Transform Failed (<status>) for context <correlator>	
	DESCRIPTION	An error has occurred while processing the CICS transform for an inbound DPL response. This normally occurs when the API entry in the gateway configuration for the local service specifies CICS instead of ATMI.
	ACTION	Check gateway configuration for incorrect specification of local service API entry.
9037:ERROR	Outbound Request Transform Failed (<status>) for context <correlator>	
	DESCRIPTION	An error has occurred while processing the CICS transform for an outbound DPL request. This normally occurs when the API entry in the gateway configuration for the remote service specifies CICS instead of ATMI.
	ACTION	Check gateway configuration for incorrect specification of local service API entry.
9038:ERROR	Outbound Response Transform Failed (<status>) for context <correlator>	
	DESCRIPTION	An error has occurred while processing the CICS transform for an outbound DPL response. This normally occurs when the API entry in the gateway configuration for the remote service specifies CICS instead of ATMI.
	ACTION	Check gateway configuration for incorrect specification of local service API entry.
9039:ERROR	Conversation terminated without confirm for context <correlator>	
	DESCRIPTION	Sync level 2 conversation was terminated with out confirm.
	ACTION	Check application program and correct.
9040:ERROR	Inbound Confirm not supported	
	DESCRIPTION	Host application is requesting an inbound confirm. This is not supported.
	ACTION	Check host application program and correct.
9041:ERROR	Inbound Confirm for multi-ISRT not supported	
	DESCRIPTION	Host IMS application is requesting an inbound confirm and using multiple ISRT commands. This is not supported.
	ACTION	Check host application program and correct.
9043:ERROR	Missing send last from host (ATMI request/response) for context <correlator>	
	DESCRIPTION	Host application did not issue send last during an outbound request/response service. The host application may have abounded.
	ACTION	Check application program and correct.
9044:ERROR	DPL program abounded with CICS code <abendcode>, program=<progrname>	
	DESCRIPTION	The specified host DPL program has abounded with the code specified.
	ACTION	None. This message is for information only.
9045:ERROR	DPL program failed with CICS code <eibrcode>, program=<progrname>	
	DESCRIPTION	The specified host DPL program has failed with the eibrcode specified.
	ACTION	None. This message is for information only.

9046:ERROR	<p>Invalid combination for Service Context <correlator>, <combination></p> <p>DESCRIPTION The specified is invalid. It will be one of the following:</p> <ol style="list-style-type: none"> 1. Sync-Level, function, and API 2. Function and API <p>ACTION Examine the gateway configuration and make corrections.</p>
9047:ERROR	<p>Sequence number error for Service Context <correlator>, sedan <seqno></p> <p>DESCRIPTION There has been a sequence number failure for the specified context. Context is out of sequence.</p> <p>ACTION Contact Oracle Customer Support</p>
9048:ERROR	<p>Invalid conversation task for Service Context <correlator>, task=<task></p> <p>DESCRIPTION The conversation has already been terminated.</p> <p>ACTION Contact Oracle Customer Support.</p>
9049:ERROR	<p>Invalid task switch for Service Context <correlator>, from <task1> to <task2></p> <p>DESCRIPTION An internal protocol violation has occurred.</p> <p>ACTION Contact Oracle Customer Support.</p>
9050:ERROR	<p>Transformer creation failed for inbound transaction <trancode></p> <p>DESCRIPTION An internal error has occurred. Possibly out of memory.</p> <p>ACTION Contact Oracle Customer Support.</p>
9051:ERROR	<p>Transformer failed for inbound transaction <trancode></p> <p>DESCRIPTION An internal error has occurred. Resource name is not present. Mainframe compatibility problem.</p> <p>ACTION Contact Oracle Customer Support.</p>
9052:WARNING	<p>Inter-task Message dropped (<verbname>), farm=<parm> From: <task1> to <task2></p> <p>DESCRIPTION An internal message between two tasks has been dropped.</p> <p>ACTION None. This message is for information only.</p>
9053:ERROR	<p>Attempt to send <nnnnn> bytes (> 32767)</p> <p>DESCRIPTION The length of a send request exceeded 32767 (including overhead).</p> <p>ACTION Check application program and correct.</p>
9054:ERROR	<p>Allocation Failure for <trancode> on <remotesysid>:<error></p> <p>DESCRIPTION An Allocation error occurred.</p> <p>ACTION The reason for the failure is described by <error>. Correct problem with configuration or application.</p>
9055:ERROR	<p>Invalid Exchange Logs GDS variable received from <remotesysid></p> <p>DESCRIPTION The log files for the CRM have been incorrectly modified.</p> <p>ACTION Run CRMLOGS to examine the CRM log file. Cold start the Tuxedo application.</p>
9056:ERROR	<p>Invalid cold start received from <remotesysid>. Unrecoverable local transactions are pending.</p> <p>DESCRIPTION Attempting to cold start host while warm starting Tuxedo.</p> <p>ACTION Run CRMLOGS to examine the CRM log file. Cold start the Tuxedo application.</p>
9057:ERROR	<p>Invalid warm start received from <remotesysid>. Unknown log name.</p> <p>DESCRIPTION The log files for the CRM have been incorrectly modified.</p>

	ACTION	Run <code>CRMLOGS</code> to examine the CRM log file. Cold start the Tuxedo application.
9058:ERROR	Invalid Compare States GDS variable received from <remotesysid>	
	DESCRIPTION	The log files for the CRM have been incorrectly modified.
	ACTION	Run <code>CRMLOGS</code> to examine the CRM log file. Cold start the Tuxedo application.
9059:ERROR	Mixed Heuristic on link <linkref> for <unitofwork> Corrector [<correlator>]	
	DESCRIPTION	One side has reported committed while the other side has reported aborted.
	ACTION	Check the <code>ULOG</code> for any additional messages.
9060:WARNING	Inbound Exchange Logs Rejected for <remotesysid>	
	DESCRIPTION	Link not configured for sync level 2.
	ACTION	None. This message is for information only.
9061:WARNING	Link <linkref> not configured for sync level 2	
	DESCRIPTION	Link specified by <linkref> is not configured for sync level 2.
	ACTION	None. This message is for information only.
9062:ERROR	Exchange Logs Rejected for <remotesysid>, Restart Type or Log Name Mismatch	
	DESCRIPTION	The log files for the CRM have been incorrectly modified.
	ACTION	Run <code>CRMLOGS</code> to examine the CRM log file. Cold start the Tuxedo application.
9063:ERROR	Exchange Logs failed with <linkref>	
	DESCRIPTION	An error occurred during the exchange logs process.
	ACTION	Run <code>CRMLOGS</code> to examine the CRM log file. Cold start the Tuxedo application.
9064:ERROR	Invalid initial syncopate received from subordinate, <text>	
	DESCRIPTION	An internal error has occurred during the commit process. XA does not support syncopates originating from subordinate members.
	ACTION	Contact Oracle Customer Support
9069 ERROR	CRM encryption setup failed	
	DESCRIPTION	An error has occurred while establishing link-level encryption with the CRM.
	ACTION	Verify that the setup of encryption on both sides of the link is correct. Verify that there are common encryption levels in the ranges specified on the process command lines. Verify that the correct encryption libraries are installed.
9072 ERROR	Attempted access by unauthorized CRM client	
	DESCRIPTION	A client has attempted to access the CRM without the proper authentication or encryption setup.
	ACTION	Verify that the client should have access to the CRM. Verify that encryption is set up correctly in both the CRM and the client, and that the correct security add-on packages are installed. Verify that the authentication file is set up correctly, and that both the CRM and the client have correct access privileges.
9073 ERROR	CRM authentication setup failed	
	DESCRIPTION	An error has occurred while authenticating a CRM client.

	ACTION	Verify that the authentication file(s) for both the CRM and the client are set up correctly, and that both have correct access privileges. Verify that the authentication file is specified correctly on both command lines.
9074 ERROR	CRM Logical Unit%s is probably inactive or unconsidered	
	DESCRIPTION	An error has occurred while connecting to the APPC stack.
	ACTION	Verify that the CRM Logical Unit name is correct and is defined and active in both the stack and VTAM configuration.
9075 WARNING	WARN: Synclevel on link %s	
	DESCRIPTION	The configured Synclevel could not be negotiated.
	ACTION	Verify that the Remote LU is configured for the correct synclevel.
9076 ERROR	CRM Logical Unit %s is probably the wrong type	
	DESCRIPTION	An error has occurred while connecting to the APPC stack.
	ACTION	Verify that the CRM Logical Unit is correctly configured.
9077 ERROR	CRM Logical Unit %s is already in use	
	DESCRIPTION	An error has occurred while connecting to the APPC stack.
	ACTION	Verify that the CRM Logical Unit is not in use by another application.
9079 ERROR	CRM client message rejected -- incompatible software/protocol version (CRM API RC=%s)	
	DESCRIPTION	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.
	ACTION	The CRM rejects the message and terminates the connection. Check the version levels of the CRM and the client to ensure compatibility.
9080 ERROR	Gateway Signon request rejected -- CRM disconnect in progress	
	DESCRIPTION	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.
	ACTION	The Signon request is rejected. When the CRM has completed disconnect processing and returned to the reset state, retry the Signon request.
9081 ERROR	Failure occurred during Rollback of %s, remote resources may not be backed out	
	DESCRIPTION	An Error occurred while performing BACKEDOUT processing. Remote resources may not be backed out.
	ACTION	Check mainframe application transaction logs to determine the cause and take the appropriate action.
9082 INFO	Unable to determine if the transaction was committed on the partner side, possible heuristic situation	
	DESCRIPTION	Failed to determine the state of the global transaction due to a partial resynchronization.
	ACTION	Examine the CRM traces for more information concerning the reason for the failure. Check the CICS logs to see the transaction status.
9085:ERROR	Create Tuxedo SSL context failed	
	DESCRIPTION	The Gateway failed to create Tuxedo SSL context.
	ACTION	Contact Oracle Customer Support.
9086: ERROR	SSL handshake failed, errno=%d [%s]	
	DESCRIPTION	The Gateway failed to shake hands with SSL peer.

9087:ERROR	ACTION	Contact Oracle Customer Support.
	Get SSL Data failed	
	DESCRIPTION	The Gateway failed to get SSL data.
9088:ERROR	ACTION	Contact Oracle Customer Support.
	SNACRM Message Buffer Invalid	
	DESCRIPTION	The Gateway failed to get crm message buffer.
9089:ERROR	ACTION	Contact Oracle Customer Support.
	Wrong SSL cipher suit range	
	DESCRIPTION	The Gateway failed to parse SSL max and min cipher suit.
9090:ERROR	ACTION	The minimum key length used in the TLS cipher suite is 128, and the maximum key length is 256.
	Set TCP option failed(%d)	
	DESCRIPTION	The Gateway could not set TCP socket option.
9091:ERROR	ACTION	Contact Oracle Customer Support.
	When environment variable TM_ALLOW_NOTLS is not set, a TLS connection must be specified	
	DESCRIPTION	The server needs TM_ALLOW_NOTLS set to use encryption other than TLS.
	ACTION	Set TM_ALLOW_NOTLS to y in order to use encryption other than TLS.

C

Code Page Translation Tables

This section explains how to modify Code Page Translation Tables and shows examples of the Code Page Translation Tables that are provided with Oracle Tuxedo Mainframe Adapter for SNA software. The files actually containing these tables are located in the `$TUXDIR/udataobj/codepage` sub-directory on your product CDROM.

- [Modifying a Code Page Translation Table](#)
- [Default Tuxedo Code Page Translation Table](#)
- [United States \(00819x00037\) Code Page Translation Table](#)
- [Germany \(00819x00273\) Code Page Translation Table](#)
- [Finland/Sweden \(00819x00278\) Code Page Translation Table](#)
- [Spain \(00819x00284\) Code Page Translation Table](#)
- [Great Britain \(00819x00285\) Code Page Translation Table](#)
- [France \(00819x00297\) Code Page Translation Table](#)
- [Belgium \(00819x00500\) Code Page Translation Table](#)
- [Portugal \(00819x00860\) Code Page Translation Table](#)
- [Italy \(00819x00280\) Code Page Translation Table](#)
- [Latin-1 – \(00819x01047\) Code Page Translation Table](#)
- [Latin-2 – \(00912x00870\) Code Page Translation Table](#)

C.1 Modifying a Code Page Translation Table

The tables provide conversions between the ASCII Latin-1 character set and representative national language EBCDIC character sets. In most cases, you do not have to modify them. Simply choose the appropriate translation table for a selected language and enter its file name in the CODEPAGE specification, as explained in *Setting Up Data Translations*.

However, if you must modify a translation table to suit your purpose, be aware of the following:

- Ensure you have character mapping information and that you know which code represents a given character. This information is available from a number of sources and is not provided in this documentation. A good source is the *IBM National Language Support Reference Manual*.
- If you modify a character code in an outbound table, you must also modify its inbound counterpart.
- Build tables from scratch is not recommended.
- The tables have a common format that contains comment lines and required lines. The format must be maintained to ensure proper table operation. Comment lines begin with the # character. **Do not alter** the following required lines:

- version (100) specifies the format of the rest of the file.
- table (256) specifies the size of the table and the min/max number of bytes composing each character code.

To modify a table, perform the following steps:

1. Open the file you want to modify with the text editor of your choice. For example:

```
edit $TUXDIR/udataobj/codepage/00819x00273
```

The text editor opens the file, in this example the translation tables for Germany (00819x00273).

2. Modify the character code in the outbound table, using the editor functions.
3. Modify the counterpart character code in the inbound table, using the editor functions.
4. Repeat Steps 2 and 3 until you have completed the modifications.
5. Save the file, using the editor functions. Be sure to give it a name other than the original. **Do not save modifications to any of the original files** provided with your product CD ROM.

 **Note:**

To use the file you modified for code page translations, make sure you specify its name using the `CODEPAGE` option in the `DM_REMOTE_DOMAINS` section of the Gateway `DMCONFIG` file.

6. Exit the editor.
- Open the file you want to modify with the text editor of your choice. For example:

```
edit $TUXDIR/udataobj/codepage/00819x00273
```

The text editor opens the file, in this example the translation tables for Germany (00819x00273).

C.2 Default Tuxedo Code Page Translation Table

```
#####
# tuxedo
#      Default Tuxedo ASCII/EBCDIC character translation tables.
#
# Local:      "TUXEDO-ASCII"
# Remote:     "TUXEDO-EBCDIC"
# Built:      1999-04-13 22:12:00 UT
#
# @(#) $Id: tuxedo,v 1.1 1999/04/16 20:08:09 david Exp $
#-----
# Header
```

```

version 100;

# Outbound (local -> remote) table
table 256 1 1;
 00 01 02 03   37 2D 2E 2F   16 05 25 0B   0C 0D 0E 0F   # 00-0F
 10 11 12 13   3C 3D 3E 26   18 19 3F 27   1C 1D 1E 1F   # 10-1F
 40 5A 7F 7B   5B 6C 50 7D   4D 5D 5C 4E   6B 60 4B 61   # 20-2F
 F0 F1 F2 F3   F4 F5 F6 F7   F8 F9 7A 5E   4C 7E 6E 6F   # 30-3F
 7C C1 C2 C3   C4 C5 C6 C7   C8 C9 D1 D2   D3 D4 D5 D6   # 40-4F
 D7 D8 D9 E2   E3 E4 E5 E6   E7 E8 E9 AD   E0 BD 5F 6D   # 50-5F
 79 81 82 83   84 85 86 87   88 89 91 92   93 94 95 96   # 60-6F
 97 98 99 A2   A3 A4 A5 A6   A7 A8 A9 C0   6A D0 A1 07   # 70-7F
 20 21 22 23   24 15 06 17   28 29 2A 2B   2C 09 0A 1B   # 80-8F
 30 31 1A 33   34 35 36 08   38 39 3A 3B   04 14 3E E1   # 90-9F
 41 42 43 44   45 46 47 48   49 51 52 53   54 55 56 57   # A0-AF
 58 59 62 63   64 65 66 67   68 69 70 71   72 73 74 75   # B0-BF
 77 78 80 8A   8B 8C 8D 8E   8F 90 9A 9B   9C 9D 9E     # C0-CF
 9F A0 AA AB   AC 4A AE AF   B0 B1 B2 B3   B4 B5 B6 B7   # D0-DF
 B8 B9 BA BB   BC 4F BE BF   CA CB CC CD   CE CF DA DB   # E0-EF
 DC DD DE DF   EA EB EC ED   EE EF FA FB   FC FD FE FF   # F0-FF

# Inbound (remote -> local) table
table 256 1 1;
 00 01 02 03   9C 09 86 7F   97 8D 8E 0B   0C 0D 0E 0F   # 00-0F
 10 11 12 13   9D 85 08 87   18 19 92 8F   1C 1D 1E 1F   # 10-1F
 80 81 82 83   84 0A 17 1B   88 89 8A 8B   8C 05 06 07   # 20-2F
 90 91 16 93   94 95 96 04   98 99 9A 9B   14 15 9E 1A   # 30-3F
 20 A0 A1 A2   A3 A4 A5 A6   A7 A8 D5 2E   3C 28 2B E5   # 40-4F
 26 A9 AA AB   AC AD AE AF   B0 B1 21 24   2A 29 3B 5E   # 50-5F
 2D 2F B2 B3   B4 B5 B6 B7   B8 B9 7C 2C   25 5F 3E 3F   # 60-6F
 BA BB BC BD   BE BF C0 C1   C2 60 3A 23   40 27 3D 22   # 70-7F
 C3 61 62 63   64 65 66 67   68 69 C4 C5   C6 C7 C8 C9   # 80-8F
 CA 6A 6B 6C   6D 6E 6F 70   71 72 CB CC   CD CE CF D0   # 90-9F
 D1 7E 73 74   75 76 77 78   79 7A D2 D3   D4 5B D6 D7   # A0-AF
 D8 D9 DA DB   DC DD DE DF   E0 E1 E2 E3   E4 5D E6 E7   # B0-BF
 7B 41 42 43   44 45 46 47   48 49 E8 E9   EA EB EC ED   # C0-CF
 7D 4A 4B 4C   4D 4E 4F 50   51 52 EE EF   F0 F1 F2 F3   # D0-DF
 5C 9F 53 54   55 56 57 58   59 5A F4 F5   F6 F7 F8 F9   # E0-EF
 30 31 32 33   34 35 36 37   38 39 FA FB   FC FD FE FF   # F0-FF

# End

```

C.3 United States (00819x00037) Code Page Translation Table

```

#=====
# 00819x00037
#Character code page mapping tables for US (USA).
#
# Local:      "IBM-CP00819", ISO-8859-1 Latin-1
# Remote:    "IBM-CP00037", EBCDIC Latin-1, US
# Built:     1999-04-16 19:50:00 UT
#
# @(#) $Id: 00819x00037,v 1.3.2.1 1999/04/29 13:03:56 cmadm Exp $

```

```

#-----
# Header
version 100;

# Outbound (local -> remote) table
table 256 1 1;
 00 01 02 03   37 2D 2E 2F   16 05 15 0B   0C 0D 0E 0F   # 00-0F
 10 11 12 13   3C 3D 32 26   18 19 3F 27   1C 1D 1E 1F   # 10-1F
 40 5A 7F 7B   5B 6C 50 7D   4D 5D 5C 4E   6B 60 4B 61   # 20-2F
 F0 F1 F2 F3   F4 F5 F6 F7   F8 F9 7A 5E   4C 7E 6E 6F   # 30-3F
 7C C1 C2 C3   C4 C5 C6 C7   C8 C9 D1 D2   D3 D4 D5 D6   # 40-4F
 D7 D8 D9 E2   E3 E4 E5 E6   E7 E8 E9 BA   E0 BB B0 6D   # 50-5F
 79 81 82 83   84 85 86 87   88 89 91 92   93 94 95 96   # 60-6F
 97 98 99 A2   A3 A4 A5 A6   A7 A8 A9 C0   4F D0 A1 07   # 70-7F
 04 06 08 09   0A 14 17 1A   1B 20 21 22   23 24 25 28   # 80-8F
 29 2A 2B 2C   30 31 33 34   35 36 38 39   3A 3B 3E FF   # 90-9F
 41 AA 4A B1   9F B2 6A B5   BD B4 9A 8A   5F CA AF BC   # A0-AF
 90 8F EA FA   BE A0 B6 B3   9D DA 9B 8B   B7 B8 B9 AB   # B0-BF
 64 65 62 66   63 67 9E 68   74 71 72 73   78 75 76 77   # C0-CF
 AC 69 ED EE   EB EF EC BF   80 FD FE FB   FC AD AE 59   # D0-DF
 44 45 42 46   43 47 9C 48   54 51 52 53   58 55 56 57   # E0-EF
 8C 49 CD CE   CB CF CC E1   70 DD DE DB   DC 8D 8E DF   # F0-FF

# Inbound (remote -> local) table
table 256 1 1;
 00 01 02 03   80 09 81 7F   82 83 84 0B   0C 0D 0E 0F   # 00-0F
 10 11 12 13   85 0A 08 86   18 19 87 88   1C 1D 1E 1F   # 10-1F
 89 8A 8B 8C   8D 8E 17 1B   8F 90 91 92   93 05 06 07   # 20-2F
 94 95 16 96   97 98 99 04   9A 9B 9C 9D   14 15 9E 1A   # 30-3F
 20 A0 E2 E4   E0 E1 E3 E5   E7 F1 A2 2E   3C 28 2B 7C   # 40-4F
 26 E9 EA EB   E8 ED EE EF   EC DF 21 24   2A 29 3B AC   # 50-5F
 2D 2F C2 C4   C0 C1 C3 C5   C7 D1 A6 2C   25 5F 3E 3F   # 60-6F
 F8 C9 CA CB   C8 CD CE CF   CC 60 3A 23   40 27 3D 22   # 70-7F
 D8 61 62 63   64 65 66 67   68 69 AB BB   F0 FD FE B1   # 80-8F
 B0 6A 6B 6C   6D 6E 6F 70   71 72 AA BA   E6 B8 C6 A4   # 90-9F
 B5 7E 73 74   75 76 77 78   79 7A A1 BF   D0 DD DE AE   # A0-AF
 5E A3 A5 B7   A9 A7 B6 BC   BD BE 5B 5D   AF A8 B4 D7   # B0-BF
 7B 41 42 43   44 45 46 47   48 49 AD F4   F6 F2 F3 F5   # C0-CF
 7D 4A 4B 4C   4D 4E 4F 50   51 52 B9 FB   FC F9 FA FF   # D0-DF
 5C F7 53 54   55 56 57 58   59 5A B2 D4   D6 D2 D3 D5   # E0-EF
 30 31 32 33   34 35 36 37   38 39 B3 DB   DC D9 DA 9F   # F0-FF

# End

```

C.4 Germany (00819x00273) Code Page Translation Table

```

#=====
# 00819x00273
#Character code page mapping tables for Germany (Deutschland).
#
# Local:   "IBM-CP00819", ISO-8859-1 Latin-1
# Remote:  "IBM-CP00273", EBCDIC Latin-1, Germany

```



```

# Built:    1999-04-16 21:00:00 UT
#
# @(#) $Id: 00819x00273,v 1.4.2.1 1999/04/29 13:04:18 cmadm Exp $
#-----
# Header
version 100;

# Outbound (local -> remote) table
table 256 1 1;
 00 01 02 03   37 2D 2E 2F   16 05 15 0B   0C 0D 0E 0F   # 00-0F
 10 11 12 13   3C 3D 32 26   18 19 3F 27   1C 1D 1E 1F   # 10-1F
 40 4F 7F 7B   5B 6C 50 7D   4D 5D 5C 4E   6B 60 4B 61   # 20-2F
 F0 F1 F2 F3   F4 F5 F6 F7   F8 F9 7A 5E   4C 7E 6E 6F   # 30-3F
 B5 C1 C2 C3   C4 C5 C6 C7   C8 C9 D1 D2   D3 D4 D5 D6   # 40-4F
 D7 D8 D9 E2   E3 E4 E5 E6   E7 E8 E9 63   EC FC 5F 6D   # 50-5F
 79 81 82 83   84 85 86 87   88 89 91 92   93 94 95 96   # 60-6F
 97 98 99 A2   A3 A4 A5 A6   A7 A8 A9 43   BB DC 59 07   # 70-7F
 04 06 08 09   0A 14 17 1A   1B 20 21 22   23 24 25 28   # 80-8F
 29 2A 2B 2C   30 31 33 34   35 36 38 39   3A 3B 3E FF   # 90-9F
 41 AA B0 B1   9F B2 CC 7C   BD B4 9A 8A   BA CA AF BC   # A0-AF
 90 8F EA FA   BE A0 B6 B3   9D DA 9B 8B   B7 B8 B9 AB   # B0-BF
 64 65 62 66   4A 67 9E 68   74 71 72 73   78 75 76 77   # C0-CF
 69 ED EE EB   EF E0 BF 80   FD FE FB 5A   AD AE A1       # D0-DF
 44 45 42 46   C0 47 9C 48   54 51 52 53   58 55 56 57   # E0-EF
 8C 49 CD CE   CB CF 6A E1   70 DD DE DB   D0 8D 8E DF   # F0-FF

# Inbound (remote -> local) table
table 256 1 1;
 00 01 02 03   80 09 81 7F   82 83 84 0B   0C 0D 0E 0F   # 00-0F
 10 11 12 13   85 0A 08 86   18 19 87 88   1C 1D 1E 1F   # 10-1F
 89 8A 8B 8C   8D 8E 17 1B   8F 90 91 92   93 05 06 07   # 20-2F
 94 95 16 96   97 98 99 04   9A 9B 9C 9D   14 15 9E 1A   # 30-3F
 20 A0 E2 7B   E0 E1 E3 E5   E7 F1 C4 2E   3C 28 2B 21   # 40-4F
 26 E9 EA EB   E8 ED EE EF   EC 7E DC 24   2A 29 3B 5E   # 50-5F
 2D 2F C2 5B   C0 C1 C3 C5   C7 D1 F6 2C   25 5F 3E 3F   # 60-6F
 F8 C9 CA CB   C8 CD CE CF   CC 60 3A 23   A7 27 3D 22   # 70-7F
 D8 61 62 63   64 65 66 67   68 69 AB BB   F0 FD FE B1   # 80-8F
 B0 6A 6B 6C   6D 6E 6F 70   71 72 AA BA   E6 B8 C6 A4   # 90-9F
 B5 DF 73 74   75 76 77 78   79 7A A1 BF   D0 DD DE AE   # A0-AF
 A2 A3 A5 B7   A9 40 B6 BC   BD BE AC 7C   AF A8 B4 D7   # B0-BF
 E4 41 42 43   44 45 46 47   48 49 AD F4   A6 F2 F3 F5   # C0-CF
 FC 4A 4B 4C   4D 4E 4F 50   51 52 B9 FB   7D F9 FA FF   # D0-DF
 D6 F7 53 54   55 56 57 58   59 5A B2 D4   5C D2 D3 D5   # E0-EF
 30 31 32 33   34 35 36 37   38 39 B3 DB   5D D9 DA 9F   # F0-FF

# End

```

C.5 Finland/Sweden (00819x00278) Code Page Translation Table

```

#=====
# 00819x00278

```

```

#Character code page mapping tables for Finland/Sweden.
#
# Local:   "IBM-CP00819", ISO-8859-1 Latin-1
# Remote:  "IBM-CP00278", EBCDIC Latin-1, Finland/Sweden
# Built:   1999-04-16 19:50:00 UT
#
# @(#) $Id: 00819x00278,v 1.4.2.1 1999/04/29 13:04:01 cmadm Exp $
#-----
# Header
version 100;

# Outbound (local -> remote) table
table 256 1 1;
 00 01 02 03   37 2D 2E 2F   16 05 15 0B   0C 0D 0E 0F   # 00-0F
 10 11 12 13   3C 3D 32 26   18 19 3F 27   1C 1D 1E 1F   # 10-1F
 40 4F 7F 63   67 6C 50 7D   4D 5D 5C 4E   6B 60 4B 61   # 20-2F
 F0 F1 F2 F3   F4 F5 F6 F7   F8 F9 7A 5E   4C 7E 6E 6F   # 30-3F
 EC C1 C2 C3   C4 C5 C6 C7   C8 C9 D1 D2   D3 D4 D5 D6   # 40-4F
 D7 D8 D9 E2   E3 E4 E5 E6   E7 E8 E9 B5   71 9F 5F 6D   # 50-5F
 51 81 82 83   84 85 86 87   88 89 91 92   93 94 95 96   # 60-6F
 97 98 99 A2   A3 A4 A5 A6   A7 A8 A9 43   BB 47 DC 07   # 70-7F
 04 06 08 09   0A 14 17 1A   1B 20 21 22   23 24 25 28   # 80-8F
 29 2A 2B 2C   30 31 33 34   35 36 38 39   3A 3B 3E FF   # 90-9F
 41 AA B0 B1   5A B2 CC 4A   BD B4 9A 8A   BA CA AF BC   # A0-AF
 90 8F EA FA   BE A0 B6 B3   9D DA 9B 8B   B7 B8 B9 AB   # B0-BF
 64 65 62 66   7B 5B 9E 68   74 E0 72 73   78 75 76 77   # C0-CF
 AC 69 ED EE   EB EF 7C BF   80 FD FE FB   FC AD AE 59   # D0-DF
 44 45 42 46   C0 D0 9C 48   54 79 52 53   58 55 56 57   # E0-EF
 8C 49 CD CE   CB CF 6A E1   70 DD DE DB   A1 8D 8E DF   # F0-FF

# Inbound (remote -> local) table
table 256 1 1;
 00 01 02 03   80 09 81 7F   82 83 84 0B   0C 0D 0E 0F   # 00-0F
 10 11 12 13   85 0A 08 86   18 19 87 88   1C 1D 1E 1F   # 10-1F
 89 8A 8B 8C   8D 8E 17 1B   8F 90 91 92   93 05 06 07   # 20-2F
 94 95 16 96   97 98 99 04   9A 9B 9C 9D   14 15 9E 1A   # 30-3F
 20 A0 E2 7B   E0 E1 E3 7D   E7 F1 A7 2E   3C 28 2B 21   # 40-4F
 26 60 EA EB   E8 ED EE EF   EC DF A4 C5   2A 29 3B 5E   # 50-5F
 2D 2F C2 23   C0 C1 C3 24   C7 D1 F6 2C   25 5F 3E 3F   # 60-6F
 F8 5C CA CB   C8 CD CE CF   CC E9 3A C4   D6 27 3D 22   # 70-7F
 D8 61 62 63   64 65 66 67   68 69 AB BB   F0 FD FE B1   # 80-8F
 B0 6A 6B 6C   6D 6E 6F 70   71 72 AA BA   E6 B8 C6 5D   # 90-9F
 B5 FC 73 74   75 76 77 78   79 7A A1 BF   D0 DD DE AE   # A0-AF
 A2 A3 A5 B7   A9 5B B6 BC   BD BE AC 7C   AF A8 B4 D7   # B0-BF
 E4 41 42 43   44 45 46 47   48 49 AD F4   A6 F2 F3 F5   # C0-CF
 E5 4A 4B 4C   4D 4E 4F 50   51 52 B9 FB   7E F9 FA FF   # D0-DF
 C9 F7 53 54   55 56 57 58   59 5A B2 D4   40 D2 D3 D5   # E0-EF
 30 31 32 33   34 35 36 37   38 39 B3 DB   DC D9 DA 9F   # F0-FF

# End

```

C.6 Spain (00819x00284) Code Page Translation Table

```
#=====
# 00819x00284
#Character code page mapping tables for Spain (España).
#
# Local:      "IBM-CP00819", ISO-8859-1 Latin-1
# Remote:    "IBM-CP00284", EBCDIC Latin-1, Spain
# Built:     1999-04-16 19:50:00 UT
#
# @(#) $Id: 00819x00284,v 1.4.2.1 1999/04/29 13:04:22 cmadm Exp $
#-----
# Header
version 100;

# Outbound (local -> remote) table
table 256 1 1;
 00 01 02 03   37 2D 2E 2F   16 05 15 0B   0C 0D 0E 0F   # 00-0F
 10 11 12 13   3C 3D 32 26   18 19 3F 27   1C 1D 1E 1F   # 10-1F
 40 BB 7F 69   5B 6C 50 7D   4D 5D 5C 4E   6B 60 4B 61   # 20-2F
 F0 F1 F2 F3   F4 F5 F6 F7   F8 F9 7A 5E   4C 7E 6E 6F   # 30-3F
 7C C1 C2 C3   C4 C5 C6 C7   C8 C9 D1 D2   D3 D4 D5 D6   # 40-4F
 D7 D8 D9 E2   E3 E4 E5 E6   E7 E8 E9 4A   E0 5A BA 6D   # 50-5F
 79 81 82 83   84 85 86 87   88 89 91 92   93 94 95 96   # 60-6F
 97 98 99 A2   A3 A4 A5 A6   A7 A8 A9 C0   4F D0 BD 07   # 70-7F
 04 06 08 09   0A 14 17 1A   1B 20 21 22   23 24 25 28   # 80-8F
 29 2A 2B 2C   30 31 33 34   35 36 38 39   3A 3B 3E FF   # 90-9F
 41 AA B0 B1   9F B2 49 B5   A1 B4 9A 8A   35F CA AF BC  # A0-AF
 90 8F EA FA   BE A0 B6 B3   9D DA 9B 8B   B7 B8 B9 AB   # B0-BF
 64 65 62 66   63 67 9E 68   74 71 72 73   78 75 76 77   # C0-CF
 AC 7B ED EE   EB EF EC BF   80 FD FE FB   FC AD AE 59   # D0-DF
 44 45 42 46   43 47 9C 48   54 51 52 53   58 55 56 57   # E0-EF
 8C 6A CD CE   CB CF CC E1   70 DD DE DB   DC 8D 8E DF   # F0-FF

# Inbound (remote -> local) table
table 256 1 1;
 00 01 02 03   80 09 81 7F   82 83 84 0B   0C 0D 0E 0F   # 00-0F
 10 11 12 13   85 0A 08 86   18 19 87 88   1C 1D 1E 1F   # 10-1F
 89 8A 8B 8C   8D 8E 17 1B   8F 90 91 92   93 05 06 07   # 20-2F
 94 95 16 96   97 98 99 04   9A 9B 9C 9D   14 15 9E 1A   # 30-3F
 20 A0 E2 E4   E0 E1 E3 E5   E7 A6 5B 2E   3C 28 2B 7C   # 40-4F
 26 E9 EA EB   E8 ED EE EF   EC DF 5D 24   2A 29 3B AC   # 50-5F
 2D 2F C2 C4   C0 C1 C3 C5   C7 23 F1 2C   25 5F 3E 3F   # 60-6F
 F8 C9 CA CB   C8 CD CE CF   CC 60 3A D1   40 27 3D 22   # 70-7F
 D8 61 62 63   64 65 66 67   68 69 AB BB   F0 FD FE B1   # 80-8F
 B0 6A 6B 6C   6D 6E 6F 70   71 72 AA BA   E6 B8 C6 A4   # 90-9F
 B5 A8 73 74   75 76 77 78   79 7A A1 BF   D0 DD DE AE   # A0-AF
 A2 A3 A5 B7   A9 A7 B6 BC   BD BE 5E 21   AF 7E B4 D7   # B0-BF
 7B 41 42 43   44 45 46 47   48 49 AD F4   F6 F2 F3 F5   # C0-CF
 7D 4A 4B 4C   4D 4E 4F 50   51 52 B9 FB   FC F9 FA FF   # D0-DF
 5C F7 53 54   55 56 57 58   59 5A B2 D4   D6 D2 D3 D5   # E0-EF
 30 31 32 33   34 35 36 37   38 39 B3 DB   DC D9 DA 9F   # F0-FF
```

End

C.7 Great Britain (00819x00285) Code Page Translation Table

```

=====
# 00819x00285
#Character code page mapping tables for Great Britain (UK).
#
# Local:      "IBM-CP00819", ISO-8859-1 Latin-1
# Remote:     "IBM-CP00285", EBCDIC Latin-1, UK
# Built:      1999-04-16 19:50:00 UT
#
# @(#) $Id: 00819x00285,v 1.5.2.1 1999/04/29 13:04:04 cmadm Exp $
#-----
# Header
version 100;

# Outbound (local -> remote) table
table 256 1 1;
  00 01 02 03   37 2D 2E 2F   16 05 15 0B   0C 0D 0E 0F   # 00-0F
  10 11 12 13   3C 3D 32 26   18 19 3F 27   1C 1D 1E 1F   # 10-1F
  40 5A 7F 7B   4A 6C 50 7D   4D 5D 5C 4E   6B 60 4B 61   # 20-2F
  F0 F1 F2 F3   F4 F5 F6 F7   F8 F9 7A 5E   4C 7E 6E 6F   # 30-3F
  7C C1 C2 C3   C4 C5 C6 C7   C8 C9 D1 D2   D3 D4 D5 D6   # 40-4F
  D7 D8 D9 E2   E3 E4 E5 E6   E7 E8 E9 B1   E0 BB BA 6D   # 50-5F
  79 81 82 83   84 85 86 87   88 89 91 92   93 94 95 96   # 60-6F
  97 98 99 A2   A3 A4 A5 A6   A7 A8 A9 C0   4F D0 BC 07   # 70-7F
  04 06 08 09   0A 14 17 1A   1B 20 21 22   23 24 25 28   # 80-8F
  29 2A 2B 2C   30 31 33 34   35 36 38 39   3A 3B 3E FF   # 90-9F
  41 AA B0 5B   9F B2 6A B5   BD B4 9A 8A   5F CA AF A1   # A0-AF
  90 8F EA FA   BE A0 B6 B3   9D DA 9B 8B   B7 B8 B9 AB   # B0-BF
  64 65 62 66   63 67 9E 68   74 71 72 73   78 75 76 77   # C0-CF
  AC 69 ED EE   EB EF EC BF   80 FD FE FB   FC AD AE 59   # D0-DF
  44 45 42 46   43 47 9C 48   54 51 52 53   58 55 56 57   # E0-EF
  8C 49 CD CE   CB CF CC E1   70 DD DE DB   DC 8D 8E DF   # F0-FF

# Inbound (remote -> local) table
table 256 1 1;
  00 01 02 03   80 09 81 7F   82 83 84 0B   0C 0D 0E 0F   # 00-0F
  10 11 12 13   85 0A 08 86   18 19 87 88   1C 1D 1E 1F   # 10-1F
  89 8A 8B 8C   8D 8E 17 1B   8F 90 91 92   93 05 06 07   # 20-2F
  94 95 16 96   97 98 99 04   9A 9B 9C 9D   14 15 9E 1A   # 30-3F
  20 A0 E2 E4   E0 E1 E3 E5   E7 F1 24 2E   3C 28 2B 7C   # 40-4F
  26 E9 EA EB   E8 ED EE EF   EC DF 21 A3   2A 29 3B AC   # 50-5F
  2D 2F C2 C4   C0 C1 C3 C5   C7 D1 A6 2C   25 5F 3E 3F   # 60-6F
  F8 C9 CA CB   C8 CD CE CF   CC 60 3A 23   40 27 3D 22   # 70-7F
  D8 61 62 63   64 65 66 67   68 69 AB BB   F0 FD FE B1   # 80-8F
  B0 6A 6B 6C   6D 6E 6F 70   71 72 AA BA   E6 B8 C6 A4   # 90-9F
  B5 AF 73 74   75 76 77 78   79 7A A1 BF   D0 DD DE AE   # A0-AF
  A2 5B A5 B7   A9 A7 B6 BC   BD BE 5E 5D   7E A8 B4 D7   # B0-BF

```

```

7B 41 42 43   44 45 46 47   48 49 AD F4   F6 F2 F3 F5   # C0-CF
7D 4A 4B 4C   4D 4E 4F 50   51 52 B9 FB   FC F9 FA FF   # D0-DF
5C F7 53 54   55 56 57 58   59 5A B2 D4   D6 D2 D3 D5   # E0-EF
30 31 32 33   34 35 36 37   38 39 B3 DB   DC D9 DA 9F   # F0-FF

```

End

C.8 France (00819x00297) Code Page Translation Table

```

#=====
# 00819x00297
#Character code page mapping tables for France.
#
# Local:      "IBM-CP00819", ISO-8859-1 Latin-1
# Remote:    "IBM-CP00297", EBCDIC Latin-1, France
# Built:     1999-04-16 23:30:00 UT
#
# @(#) $Id: 00819x00297,v 1.4.2.1 1999/04/29 13:04:27 cmadm Exp $
#-----
# Header
version 100;

# Outbound (local -> remote) table
table 256 1 1;
 00 01 02 03   37 2D 2E 2F   16 05 15 0B   0C 0D 0E 0F   # 00-0F
 10 11 12 13   3C 3D 32 26   18 19 3F 27   1C 1D 1E 1F   # 10-1F
 40 4F 7F B1   5B 6C 50 7D   4D 5D 5C 4E   6B 60 4B 61   # 20-2F
 F0 F1 F2 F3   F4 F5 F6 F7   F8 F9 7A 5E   4C 7E 6E 6F   # 30-3F
 44 C1 C2 C3   C4 C5 C6 C7   C8 C9 D1 D2   D3 D4 D5 D6   # 40-4F
 D7 D8 D9 E2   E3 E4 E5 E6   E7 E8 E9 90   48 B5 5F 6D   # 50-5F
 A0 81 82 83   84 85 86 87   88 89 91 92   93 94 95 96   # 60-6F
 97 98 99 A2   A3 A4 A5 A6   A7 A8 A9 51   BB 54 BD 07   # 70-7F
 04 06 08 09   0A 14 17 1A   1B 20 21 22   23 24 25 28   # 80-8F
 29 2A 2B 2C   30 31 33 34   35 36 38 39   3A 3B 3E FF   # 90-9F
 41 AA B0 7B   9F B2 DD 5A   A1 B4 9A 8A   BA CA AF BC   # A0-AF
 4A 8F EA FA   BE 79 B6 B3   9D DA 9B 8B   B7 B8 B9 AB   # B0-BF
 64 65 62 66   63 67 9E 68   74 71 72 73   78 75 76 77   # C0-CF
 AC 69 ED EE   EB EF EC BF   80 FD FE FB   FC AD AE 59   # D0-DF
 7C 45 42 46   43 47 9C E0   D0 C0 52 53   58 55 56 57   # E0-EF
 8C 49 CD CE   CB CF CC E1   70 6A DE DB   DC 8D 8E DF   # F0-FF

# Inbound (remote -> local) table
table 256 1 1;
 00 01 02 03   80 09 81 7F   82 83 84 0B   0C 0D 0E 0F   # 00-0F
 10 11 12 13   85 0A 08 86   18 19 87 88   1C 1D 1E 1F   # 10-1F
 89 8A 8B 8C   8D 8E 17 1B   8F 90 91 92   93 05 06 07   # 20-2F
 94 95 16 96   97 98 99 04   9A 9B 9C 9D   14 15 9E 1A   # 30-3F
 20 A0 E2 E4   40 E1 E3 E5   5C F1 B0 2E   3C 28 2B 21   # 40-4F
 26 7B EA EB   7D ED EE EF   EC DF A7 24   2A 29 3B 5E   # 50-5F
 2D 2F C2 C4   C0 C1 C3 C5   C7 D1 F9 2C   25 5F 3E 3F   # 60-6F
 F8 C9 CA CB   C8 CD CE CF   CC B5 3A A3   E0 27 3D 22   # 70-7F
 D8 61 62 63   64 65 66 67   68 69 AB BB   F0 FD FE B1   # 80-8F
 5B 6A 6B 6C   6D 6E 6F 70   71 72 AA BA   E6 B8 C6 A4   # 90-9F

```

```

60 A8 73 74   75 76 77 78   79 7A A1 BF   D0 DD DE AE   # A0-AF
A2 23 A5 B7   A9 5D B6 BC   BD BE AC 7C   AF 7E B4 D7   # B0-BF
E9 41 42 43   44 45 46 47   48 49 AD F4   F6 F2 F3 F5   # C0-CF
E8 4A 4B 4C   4D 4E 4F 50   51 52 B9 FB   FC A6 FA FF   # D0-DF
E7 F7 53 54   55 56 57 58   59 5A B2 D4   D6 D2 D3 D5   # E0-EF
30 31 32 33   34 35 36 37   38 39 B3 DB   DC D9 DA 9F   # F0-FF

```

```
# End
```

C.9 Belgium (00819x00500) Code Page Translation Table

```

#=====
# 00819x00500
#Character code page mapping tables for Belgium (Belgique).
#
# Local:   "IBM-CP00819", ISO-8859-1 Latin-1
# Remote:  "IBM-CP00500", EBCDIC Latin-1, Belgium
# Built:   1999-04-16 19:50:00 UT
#
# @(#) $Id: 00819x00500,v 1.4.2.1 1999/04/29 13:04:09 cmadm Exp $
#-----
# Header
version 100;

# Outbound (local -> remote) table
table 256 1 1;
  00 01 02 03   37 2D 2E 2F   16 05 15 0B   0C 0D 0E 0F   # 00-0F
  10 11 12 13   3C 3D 32 26   18 19 3F 27   1C 1D 1E 1F   # 10-1F
  40 4F 7F 7B   5B 6C 50 7D   4D 5D 5C 4E   6B 60 4B 61   # 20-2F
  F0 F1 F2 F3   F4 F5 F6 F7   F8 F9 7A 5E   4C 7E 6E 6F   # 30-3F
  7C C1 C2 C3   C4 C5 C6 C7   C8 C9 D1 D2   D3 D4 D5 D6   # 40-4F
  D7 D8 D9 E2   E3 E4 E5 E6   E7 E8 E9 4A   E0 5A 5F 6D   # 50-5F
  79 81 82 83   84 85 86 87   88 89 91 92   93 94 95 96   # 60-6F
  97 98 99 A2   A3 A4 A5 A6   A7 A8 A9 C0   BB D0 A1 07   # 70-7F
  04 06 08 09   0A 14 17 1A   1B 20 21 22   23 24 25 28   # 80-8F
  29 2A 2B 2C   30 31 33 34   35 36 38 39   3A 3B 3E FF   # 90-9F
  41 AA B0 B1   9F B2 6A B5   BD B4 9A 8A   BA CA AF BC   # A0-AF
  90 8F EA FA   BE A0 B6 B3   9D DA 9B 8B   B7 B8 B9 AB   # B0-BF
  64 65 62 66   63 67 9E 68   74 71 72 73   78 75 76 77   # C0-CF
  AC 69 ED EE   EB EF EC BF   80 FD FE FB   FC AD AE 59   # D0-DF
  44 45 42 46   43 47 9C 48   54 51 52 53   58 55 56 57   # E0-EF
  8C 49 CD CE   CB CF CC E1   70 DD DE DB   DC 8D 8E DF   # F0-FF

# Inbound (remote -> local) table
table 256 1 1;
  00 01 02 03   80 09 81 7F   82 83 84 0B   0C 0D 0E 0F   # 00-0F
  10 11 12 13   85 0A 08 86   18 19 87 88   1C 1D 1E 1F   # 10-1F
  89 8A 8B 8C   8D 8E 17 1B   8F 90 91 92   93 05 06 07   # 20-2F
  94 95 16 96   97 98 99 04   9A 9B 9C 9D   14 15 9E 1A   # 30-3F
  20 A0 E2 E4   E0 E1 E3 E5   E7 F1 5B 2E   3C 28 2B 21   # 40-4F
  26 E9 EA EB   E8 ED EE EF   EC DF 5D 24   2A 29 3B 5E   # 50-5F
  2D 2F C2 C4   C0 C1 C3 C5   C7 D1 A6 2C   25 5F 3E 3F   # 60-6F
  F8 C9 CA CB   C8 CD CE CF   CC 60 3A 23   40 27 3D 22   # 70-7F

```

```

D8 61 62 63   64 65 66 67   68 69 AB BB   F0 FD FE B1   # 80-8F
B0 6A 6B 6C   6D 6E 6F 70   71 72 AA BA   E6 B8 C6 A4   # 90-9F
B5 7E 73 74   75 76 77 78   79 7A A1 BF   D0 DD DE AE   # A0-AF
A2 A3 A5 B7   A9 A7 B6 BC   BD BE AC 7C   AF A8 B4 D7   # B0-BF
7B 41 42 43   44 45 46 47   48 49 AD F4   F6 F2 F3 F5   # C0-CF
7D 4A 4B 4C   4D 4E 4F 50   51 52 B9 FB   FC F9 FA FF   # D0-DF
5C F7 53 54   55 56 57 58   59 5A B2 D4   D6 D2 D3 D5   # E0-EF
30 31 32 33   34 35 36 37   38 39 B3 DB   DC D9 DA 9F   # F0-FF

```

```
# End
```

C.10 Portugal (00819x00860) Code Page Translation Table

```

=====
# 00819x00860
#Character code page mapping tables for Portugal.
#
# Local:      "IBM-CP00819", ISO-8859-1 Latin-1
# Remote:    "IBM-CP00860", ASCII IBM-PC graphics, Portugal
# Built:     1999-04-20 00:03:00 UT
#
# Caveats
#   The mapping between the two code pages is inexact, because some
#   characters do not exist in both code sets.
#
# @(#) $Id: 00819x00860,v 1.4 1999/04/20 20:19:20 david Exp $
#-----

# Header
version 100;

# Outbound (local -> remote) table
table 256 1 1;
  00 01 02 03   04 05 06 07   08 09 0A 0B   0C 0D 0E 0F   # 00-0F
  10 11 12 13   9E B0 16 17   18 19 1A 1B   1C 1D 1E 1F   # 10-1F
  20 21 22 23   24 25 26 27   28 29 2A 2B   2C 2D 2E 2F   # 20-2F
  30 31 32 33   34 35 36 37   38 39 3A 3B   3C 3D 3E 3F   # 30-3F
  40 41 42 43   44 45 46 47   48 49 4A 4B   4C 4D 4E 4F   # 40-4F
  50 51 52 53   54 55 56 57   58 59 5A 5B   5C 5D 5E 5F   # 50-5F
  60 61 62 63   64 65 66 67   68 69 6A 6B   6C 6D 6E 6F   # 60-6F
  70 71 72 73   74 75 76 77   78 79 7A 7B   7C 7D 7E 7F   # 70-7F
  B1 B2 B3 B4   B5 B6 B7 B8   B9 BA BB BC   BD BE BF C0   # 80-8F
  C1 C2 C3 C4   C5 C6 C7 C8   C9 CA CB CC   CD CE CF D0   # 90-9F
  FF AD 9B 9C   D1 D2 D3 15   D4 D5 A6 AE   AA D6 D7 D8   # A0-AF
  F8 F1 FD D9   DA E6 14 FA   DB DC A7 AF   AC AB DD A8   # B0-BF
  91 86 8F 8E   DE DF E0 80   92 90 89 E2   98 8B E3 E4   # C0-CF
  E5 A5 A9 9F   8C 99 E7 E9   E8 9D 96 EA   9A EC EE E1   # D0-DF
  85 A0 83 84   EF F0 F2 87   8A 82 88 F3   8D A1 F4 F5   # E0-EF
  EB A4 95 A2   93 94 F7 F6   ED 97 A3 F9   81 FB FC FE   # F0-FF

# Inbound (remote -> local) table
table 256 1 1;
  00 01 02 03   04 05 06 07   08 09 0A 0B   0C 0D 0E 0F   # 00-0F

```

```

10 11 12 13   B6 A7 16 17   18 19 1A 1B   1C 1D 1E 1F   # 10-1F
20 21 22 23   24 25 26 27   28 29 2A 2B   2C 2D 2E 2F   # 20-2F
30 31 32 33   34 35 36 37   38 39 3A 3B   3C 3D 3E 3F   # 30-3F
40 41 42 43   44 45 46 47   48 49 4A 4B   4C 4D 4E 4F   # 40-4F
50 51 52 53   54 55 56 57   58 59 5A 5B   5C 5D 5E 5F   # 50-5F
60 61 62 63   64 65 66 67   68 69 6A 6B   6C 6D 6E 6F   # 60-6F
70 71 72 73   74 75 76 77   78 79 7A 7B   7C 7D 7E 7F   # 70-7F
C7 FC E9 E2   E3 E0 C1 E7   EA CA E8 CD   D4 EC C3 C2   # 80-8F
C9 C0 C8 F4   F5 F2 DA F9   CC D5 DC A2   A3 D9 14 D3   # 90-9F
E1 ED F3 FA   F1 D1 AA BA   BF D2 AC BD   BC A1 AB BB   # A0-AF
15 80 81 82   83 84 85 86   87 88 89 8A   8B 8C 8D 8E   # B0-BF
8F 90 91 92   93 94 95 96   97 98 99 9A   9B 9C 9D 9E   # C0-CF
9F A4 A5 A6   A8 A9 AD AE   AF B3 B4 B8   B9 BE C4 C5   # D0-DF
C6 DF CB CE   CF D0 B5 D6   D8 D7 DB F0   DD F8 DE E4   # E0-EF
E5 B1 E6 EB   EE EF F7 F6   B0 FB B7 FD   FE B2 FF A0   # F0-FF

# End

```

C.11 Italy (00819x00280) Code Page Translation Table

```

#=====
#
# 00819x00280
#Character code page mapping tables for Italy.
#
# Local:      "IBM-CP00819", ISO-8859-1 Latin-1
# Remote:    "IBM-CP00280", IBM EBCDIC Latin-1, Italy
# Built:     2014-03-19 13:30 UT
#
# @(#) $Id: 00819x00280,v 1.1 2014/03/19 08:58:32 gumeng Exp $
#-----
#
# Header
version 100;

# Outbound (local -> remote) table
table 256 1 1;
 00 01 02 03   37 2D 2E 2F 16   05 25 0B 0C   0D 0E 0F   # 00-0F
 10 11 12 13   3C 3D 32 26 18   19 3F 27 1C   1D 1E 1F   # 10-1F
 40 4F 7F B1   5B 6C 50 7D 4D   5D 5C 4E 6B   60 4B 61   # 20-2F
 F0 F1 F2 F3   F4 F5 F6 F7 F8   F9 7A 5E 4C   7E 6E 6F   # 30-3F
 B5 C1 C2 C3   C4 C5 C6 C7 C8   C9 D1 D2 D3   D4 D5 D6   # 40-4F
 D7 D8 D9 E2   E3 E4 E5 E6 E7   E8 E9 90 48   51 5F 6D   # 50-5F
 DD 81 82 83   84 85 86 87 88   89 91 92 93   94 95 96   # 60-6F
 97 98 99 A2   A3 A4 A5 A6 A7   A8 A9 44 BB   54 58 07   # 70-7F
 20 21 22 23   24 15 06 17 28   29 2A 2B 2C   09 0A 1B   # 80-8F
 30 31 1A 33   34 35 36 08 38   39 3A 3B 04   14 3E FF   # 90-9F
 41 AA B0 7B   9F B2 CD 7C BD   B4 9A 8A BA   CA AF BC   # A0-AF
 4A 8F EA FA   BE A0 B6 B3 9D   DA 9B 8B B7   B8 B9 AB   # B0-BF
 64 65 62 66   63 67 9E 68 74   71 72 73 78   75 76 77   # C0-CF
 AC 69 ED EE   EB EF EC BF 80   FD FE FB FC   AD AE 59   # D0-DF
 C0 45 42 46   43 47 9C E0 D0   5A 52 53 A1   55 56 57   # E0-EF

```



```

      8C 49 6A CE    CB CF CC E1 70    79 DE DB DC    8D 8E DF    # F0-FF

# Inbound (remote -> local) table
table 256 1 1;
  00 01 02 03    9C 09 86 7F    97 8D 8E 0B    0C 0D 0E 0F    # 00-0F
  10 11 12 13    9D 85 08 87    18 19 92 8F    1C 1D 1E 1F    # 10-1F
  80 81 82 83    84 0A 17 1B    88 89 8A 8B    8C 05 06 07    # 20-2F
  90 91 16 93    94 95 96 04    98 99 9A 9B    14 15 9E 1A    # 30-3F
  20 A0 E2 E4    7B E1 E3 E5    5C F1 B0 2E    3C 28 2B 21    # 40-4F
  26 5D EA EB    7D ED EE EF    7E DF E9 24    2A 29 3B 5E    # 50-5F
  2D 2F C2 C4    C0 C1 C3 C5    C7 D1 F2 2C    25 5F 3E 3F    # 60-6F
  F8 C9 CA CB    C8 CD CE CF    CC F9 3A A3    A7 27 3D 22    # 70-7F
  D8 61 62 63    64 65 66 67    68 69 AB BB    F0 FD FE B1    # 80-8F
  5B 6A 6B 6C    6D 6E 6F 70    71 72 AA BA    E6 B8 C6 A4    # 90-9F
  B5 EC 73 74    75 76 77 78    79 7A A1 BF    D0 DD DE AE    # A0-AF
  A2 23 A5 B7    A9 40 B6 BC    BD BE AC 7C    AF A8 B4 D7    # B0-BF
  E0 41 42 43    44 45 46 47    48 49 AD F4    F6 A6 F3 F5    # C0-CF
  E8 4A 4B 4C    4D 4E 4F 50    51 52 B9 FB    FC 60 FA FF    # D0-DF
  E7 F7 53 54    55 56 57 58    59 5A B2 D4    D6 D2 D3 D5    # E0-EF
  30 31 32 33    34 35 36 37    38 39 B3 DB    DC D9 DA 9F    # F0-FF

# End

```

C.12 Latin-1 – (00819x01047) Code Page Translation Table

```

#=====
# 00819x01047
#Character code page mapping tables.
#
# Local:      "IBM-CP00819", ISO-8859-1 Latin-1
# Remote:    "IBM-CP01047", EBCDIC Latin-1
# Built:     1999-04-22 23:40:00 UT
#
# @(#) $Id: 00819x01047,v 1.1.2.1 1999/04/29 13:04:13 cmadm Exp $
#-----
# Header

version 100;

# Outbound (local -> remote) table
table 256 1 1;
  00 01 02 03    37 2D 2E 2F    16 05 15 0B    0C 0D 0E 0F    # 00-0F
  10 11 12 13    3C 3D 32 26    18 19 3F 27    1C 1D 1E 1F    # 10-1F
  40 5A 7F 7B    5B 6C 50 7D    4D 5D 5C 4E    6B 60 4B 61    # 20-2F
  F0 F1 F2 F3    F4 F5 F6 F7    F8 F9 7A 5E    4C 7E 6E 6F    # 30-3F
  7C C1 C2 C3    C4 C5 C6 C7    C8 C9 D1 D2    D3 D4 D5 D6    # 40-4F
  D7 D8 D9 E2    E3 E4 E5 E6    E7 E8 E9 AD    E0 BD 5F 6D    # 50-5F
  79 81 82 83    84 85 86 87    88 89 91 92    93 94 95 96    # 60-6F
  97 98 99 A2    A3 A4 A5 A6    A7 A8 A9 C0    4F D0 A1 07    # 70-7F
  04 06 08 09    0A 14 17 1A    1B 20 21 22    23 24 25 28    # 80-8F
  29 2A 2B 2C    30 31 33 34    35 36 38 39    3A 3B 3E FF    # 90-9F
  41 AA 4A B1    9F B2 6A B5    BB B4 9A 8A    B0 CA AF BC    # A0-AF
  90 8F EA FA    BE A0 B6 B3    9D DA 9B 8B    B7 B8 B9 AB    # B0-BF

```

```

64 65 62 66    63 67 9E 68    74 71 72 73    78 75 76 77    # C0-CF
AC 69 ED EE    EB EF EC BF    80 FD FE FB    FC BA AE 59    # D0-DF
44 45 42 46    43 47 9C 48    54 51 52 53    58 55 56 57    # E0-EF
8C 49 CD CE    CB CF CC E1    70 DD DE DB    DC 8D 8E DF    # F0-FF

# Inbound (remote -> local) table
table 256 1 1;
00 01 02 03    80 09 81 7F    82 83 84 0B    0C 0D 0E 0F    # 00-0F
10 11 12 13    85 0A 08 86    18 19 87 88    1C 1D 1E 1F    # 10-1F
89 8A 8B 8C    8D 8E 17 1B    8F 90 91 92    93 05 06 07    # 20-2F
94 95 16 96    97 98 99 04    9A 9B 9C 9D    14 15 9E 1A    # 30-3F
20 A0 E2 E4    E0 E1 E3 E5    E7 F1 A2 2E    3C 28 2B 7C    # 40-4F
26 E9 EA EB    E8 ED EE EF    EC DF 21 24    2A 29 3B 5E    # 50-5F
2D 2F C2 C4    C0 C1 C3 C5    C7 D1 A6 2C    25 5F 3E 3F    # 60-6F
F8 C9 CA CB    C8 CD CE CF    CC 60 3A 23    40 27 3D 22    # 70-7F
D8 61 62 63    64 65 66 67    68 69 AB BB    F0 FD FE B1    # 80-8F
B0 6A 6B 6C    6D 6E 6F 70    71 72 AA BA    E6 B8 C6 A4    # 90-9F
B5 7E 73 74    75 76 77 78    79 7A A1 BF    D0 5B DE AE    # A0-AF
AC A3 A5 B7    A9 A7 B6 BC    BD BE DD A8    AF 5D B4 D7    # B0-BF
7B 41 42 43    44 45 46 47    48 49 AD F4    F6 F2 F3 F5    # C0-CF
7D 4A 4B 4C    4D 4E 4F 50    51 52 B9 FB    FC F9 FA FF    # D0-DF
5C F7 53 54    55 56 57 58    59 5A B2 D4    D6 D2 D3 D5    # E0-EF
30 31 32 33    34 35 36 37    38 39 B3 DB    DC D9 DA 9F    # F0-FF

# End

```

C.13 Latin-2 – (00912x00870) Code Page Translation Table

```

#=====
# 00912x00870
#Character code page mapping tables for Latin-2 character sets.
#
# Local:      "IBM-CP00912", ISO-8859-2 Latin-2
# Remote:     "IBM-CP00870", EBCDIC Latin-2
# Built:      1999-04-16 19:50:00 UT
#
# @(#) $Id: 00912x00870,v 1.3.2.1 1999/04/29 13:04:32 cmadm Exp $
#-----

# Header
version 100;

# Outbound (local -> remote) table
table 256 1 1;
00 01 02 03    37 2D 2E 2F    16 05 15 0B    0C 0D 0E 0F    # 00-0F
10 11 12 13    3C 3D 32 26    18 19 3F 27    1C 1D 1E 1F    # 10-1F
40 4F 7F 7B    5B 6C 50 7D    4D 5D 5C 4E    6B 60 4B 61    # 20-2F
F0 F1 F2 F3    F4 F5 F6 F7    F8 F9 7A 5E    4C 7E 6E 6F    # 30-3F
7C C1 C2 C3    C4 C5 C6 C7    C8 C9 D1 D2    D3 D4 D5 D6    # 40-4F
D7 D8 D9 E2    E3 E4 E5 E6    E7 E8 E9 4A    E0 5A 5F 6D    # 50-5F
79 81 82 83    84 85 86 87    88 89 91 92    93 94 95 96    # 60-6F
97 98 99 A2    A3 A4 A5 A6    A7 A8 A9 C0    6A D0 A1 07    # 70-7F
04 06 08 09    0A 14 17 1A    1B 20 21 22    23 24 25 28    # 80-8F

```

```

29 2A 2B 2C   30 31 33 34   35 36 38 39   3A 3B 3E FF   # 90-9F
41 B1 80 BA   9F 77 AA B5   BD BC AF FD   B9 CA B8 B4   # A0-AF
90 A0 9E 9A   BE 57 8A 70   9D 9C 8F DD   B7 64 B6 B2   # B0-BF
ED 65 62 66   63 78 69 68   67 71 72 73   DA 75 76 FA   # C0-CF
AC BB AB EE   EB EF EC BF   AE 74 FE FB   FC AD B3 59   # D0-DF
CD 45 42 46   43 58 49 48   47 51 52 53   DF 55 56 EA   # E0-EF
8C 9B 8B CE   CB CF CC E1   8E 54 DE DB   DC 8D 44 B0   # F0-FF

```

Inbound (remote -> local) table

table 256 1 1;

```

00 01 02 03   80 09 81 7F   82 83 84 0B   0C 0D 0E 0F   # 00-0F
10 11 12 13   85 0A 08 86   18 19 87 88   1C 1D 1E 1F   # 10-1F
89 8A 8B 8C   8D 8E 17 1B   8F 90 91 92   93 05 06 07   # 20-2F
94 95 16 96   97 98 99 04   9A 9B 9C 9D   14 15 9E 1A   # 30-3F
20 A0 E2 E4   FE E1 E3 E8   E7 E6 5B 2E   3C 28 2B 21   # 40-4F
26 E9 EA EB   F9 ED EE B5   E5 DF 5D 24   2A 29 3B 5E   # 50-5F
2D 2F C2 C4   BD C1 C3 C8   C7 C6 7C 2C   25 5F 3E 3F   # 60-6F
B7 C9 CA CB   D9 CD CE A5   C5 60 3A 23   40 27 3D 22   # 70-7F
A2 61 62 63   64 65 66 67   68 69 B6 F2   F0 FD F8 BA   # 80-8F
B0 6A 6B 6C   6D 6E 6F 70   71 72 B3 F1   B9 B8 B2 A4   # 90-9F
B1 7E 73 74   75 76 77 78   79 7A A6 D2   D0 DD D8 AA   # A0-AF
FF A1 BF DE   AF A7 BE BC   AE AC A3 D1   A9 A8 B4 D7   # B0-BF
7B 41 42 43   44 45 46 47   48 49 AD F4   F6 E0 F3 F5   # C0-CF
7D 4A 4B 4C   4D 4E 4F 50   51 52 CC FB   FC BB FA EC   # D0-DF
5C F7 53 54   55 56 57 58   59 5A EF D4   D6 C0 D3 D5   # E0-EF
30 31 32 33   34 35 36 37   38 39 CF DB   DC AB DA 9F   # F0-FF

```

End

Glossary

Index