Administrator's Guide

iPlanet Application Server Enterprise Connector for CICS

Version 6.5

806-5503-02
August 2002

Copyright © 2002 Sun Microsystems, Inc., 901 San Antonio Road, Palo Alto, California 94303, U.S.A. All rights reserved.

Sun Microsystems, Inc. has intellectual property rights relating to technology embodied in the product that is described in this document. In particular, andwithout limitation, these intellectual property rights may include one or more of the U.S. patents listed at http://www.sun.com/patents and one or more additional patents or pending patent applications in the U.S. and in other countries.

This document and the product to which it pertains are distributed under licenses restricting their use, copying, distribution, and decompilation. No part of the product or of this document may be reproduced in any form by any means without prior written authorization of Sun and its licensors, if any.

Third-party software, including font technology, is copyrighted and licensed from Sun suppliers. Sun, Sun Microsystems, the Sun logo, Java, iPlanet and the iPlanet logo are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon architecture developed by Sun Microsystems,Inc. UNIX is a registered trademark in the U.S. and other countries, exclusively licensed through X/Open Company, Ltd. Federal Acquisitions: Commercial Software - Government Users Subject to Standard License Terms and Conditions.

DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

Copyright © 2002 Sun Microsystems, Inc., 901 San Antonio Road, Palo Alto, California 94303, Etats-Unis. Tous droits réservés.

Sun Microsystems, Inc. a les droits de propriété intellectuels relatants à la technologie incorporée dans le produit qui est décrit dans ce document. En particulier, et sans la limitation, ces droits de propriété intellectuels peuventinclure un ou plus des brevets américains énumérés à http://www.sun.com/patents et un ou les brevets plus supplémentaires ou les applications de brevet en attente dans les Etats - Unis et dans les autres pays.

Ce produit ou document est protégé par un copyright et distribué avec des licences qui en restreignent l'utilisation, la copie, la distribution, et la décompilation. Aucune partie de ce produit ou document ne peut être reproduite sous aucune forme, parquelque moyen que ce soit, sans l'autorisation préalable et écrite de Sun et de ses bailleurs de licence, s'il y ena.

Le logiciel détenu par des tiers, et qui comprend la technologie relative aux polices de caractères, est protégé par un copyright et licencié par des fournisseurs de Sun. Sun, Sun Microsystems, le logo Sun, Java, iPlanet et le logo iPlanet sont desmarques de fabrique ou des marques déposées de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays.

Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. Les produits protant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc. UNIX est une marque déposée aux Etats-Unis et dans d'autres pays et licenciée exlusivement par X/Open Company, Ltd.

LA DOCUMENTATION EST FOURNIE "EN L'ÉTAT" ET TOUTES AUTRES CONDITIONS, DECLARATIONS ET GARANTIES EXPRESSES OU TACITES SONT FORMELLEMENT EXCLUES, DANS LA MESURE AUTORISEE PAR LA LOI APPLICABLE, Y COMPRIS NOTAMMENT TOUTE GARANTIE IMPLICITE RELATIVE A LA QUALITE MARCHANDE, A L'APTITUDE A UNE UTILISATION PARTICULIERE OU A L'ABSENCE DE CONTREFAÇON.

Contents

List of Figures

List Of Tables

List of Procedures

List of Code Examples

Preface

The iPlanet Application Server Enterprise Connector for CICS explains how to get started with, install, manage, and configure the iPlanet Application Server Enterprise Connector for CICS.

This preface contains information about the following topics:

- Prerequisites
- Supported Platforms
- System Requirements
- What's in This Guide
- Documentation Conventions
- Online Guide
- Related Information

Prerequisites

The *iPlanet Application Server Enterprise Connector for CICS Administrator's Guide* is written for system administrators who want to install, manage and configure the CICS connector.

This guide assumes you are familiar with the following topics:

- iPlanet Application Server Programming Concepts.

  See the *iPlanet Application Server Enterprise Connector for CICS Developer's Guide* for more details.

- The Internet and the World Wide Web
- CICS and MVS Programming Concepts
- Java Programming Language

Supported Platforms

The iPlanet Application Server Enterprise Connector for CICS supports the following platforms:

Table 1 Platform Specifications

Vendor	Platform
Microsoft	Windows NT version 4.0 SP5 or Windows2000
Sun	SPARC running Solaris 2.6 or Solaris 2.8

System Requirements

To install the iPlanet Application Server for CICS, your system must have the following:

- iPlanet Application Server Version 6.5 installed
- iPlanet Unified Integration Framework (UIF) Version 6.5 installed
- Available disk space: 200 MB for Windows NT/2000, 400 MB for Solaris
- Memory per CPU: 256MB minimum; 512MB recommended
- IBM CICS running on OS/390. MVS 5.22 and CICS 3.2 and above with current APARS.

What's in This Guide

The *iPlanet Application Server Enterprise Connector for CICS Administrator's Guide* provides the information you need to understand, set up, and administer all aspects of the iPlanet Application Server Enterprise Connector for CICS.

The following table lists a short summary of what each chapter covers.

See this chapter:	If you want to do this:
Chapter 1, "Overview"	Familiarize yourself with general concepts of the iPlanet Application Server Enterprise Connector for CICS.
Chapter 2, "Installation"	Install the connector on Winodws NT/2000 or Solaris.

See this chapter:	If you want to do this:
Chapter 3, “Installing the Mainframe Component”	Install the mainframe component of the CICS connector.
Chapter 4, “Managing Data”	Manage the data sources and data objects.
Chapter 5, “Configuring the Application“	Configure the system and pooling settings.
Appendix A, “Operating the TELCO Sample“	Install the TELCO Customer Service sample, based on CICS and DB2 for OS/390.
Appendix B, “Security Exit for Mainframe Listener	Provide security to the mainframe Listener.

Documentation Conventions

File and directory paths are given in Windows format with backslashes separating directory names. For Solaris versions, the directory paths are the same, except slashes should be substituted in place of backslashes.

This guide uses URLs of the form:

```
http://server.domain/path/file.html
```

In these URLs, *server* is the name of the server on which you run your application; *domain* is your Internet domain name; *path* is the directory structure on the server; and *file* is an individual filename. Italics items in URLs are placeholders.

This guide uses the following font conventions:

- The `monospace` font is used for sample code and code listings, API and language elements (such as function names and class names), file names, pathnames, directory names, and HTML tags.
- *Italic* type is used for book titles, emphasis, variables and placeholders, and words used in the literal sense.

Online Guide

You can find the *iPlanet Application Server Enterprise Connector for CICS Administrator's Guide* online in PDF and HTML formats. To locate these files, use the following URL:

`http://docs.iplanet.com/docs/manuals/`

Related Information

In addition to this guide, there is additional information available for administrators, end users, and developers. The following lists these documents:

- *iPlanet Application Server Enterprise Connector for CICS Developer's Guide*
- *iPlanet Web Server Developer's Guide*
- *iPlanet Application Server Administrator's Guide*
- *iPlanet Application Server Installation Guide*
- *iPlanet Application Server Overview Guide*
- *iPlanet Application Server Release Notes*
- *iPlanet Application Server Administrator's Guide*
- *iPlanet Application Builder User's Guide*
- *iPlanet Application Builder Installation Guide*
- *iPlanet Application Builder Release Notes*
- *Unified Integration Framework (UIF) Release Notes*
- *Unified Integration Framework (UIF) Developer's Guide*

Chapter 1

Overview

The iPlanet Application Server Enterprise Connector for CICS is used for building and delivering scalable applications that integrate the application server with legacy CICS applications. The iPlanet Application Server Enterprise Connector for CICS enables communication between an end user and a remote CICS back-end system. This chapter introduces the three-tier, web-based computing model and describes the basic connector concepts.

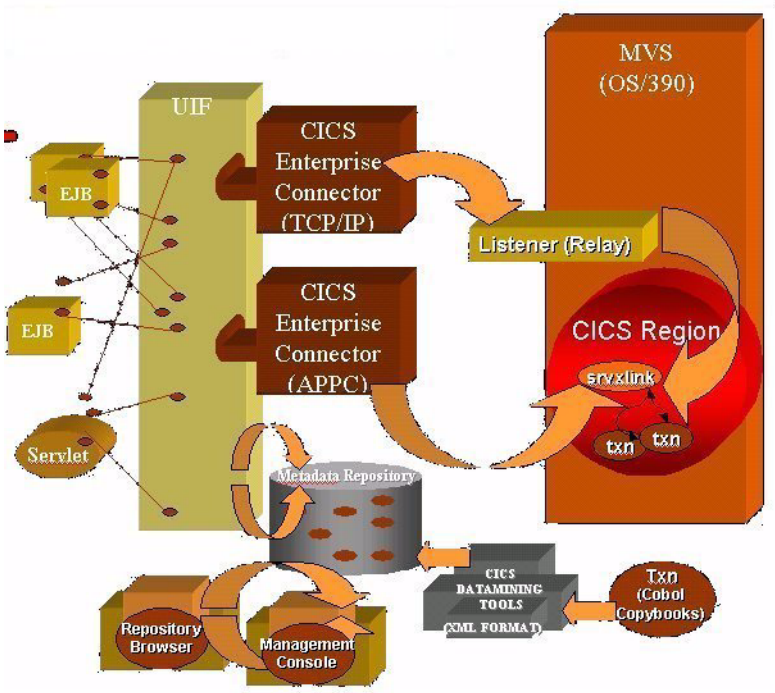
This chapter contains information about the following topics:

- Unified Integration Framework (UIF)
- Three-tier Application Model
- Enterprise Connector Tools for CICS
- National Language Support

Unified Integration Framework (UIF)

The UIF is an application programming framework that provides a single Application Programming Interface (API) to access different back-end systems. A connector is developed for each back-end system to allow communication between the UIF API and the back-end system, see Figure 1-1. The UIF API is the only API necessary to access the back-end system.

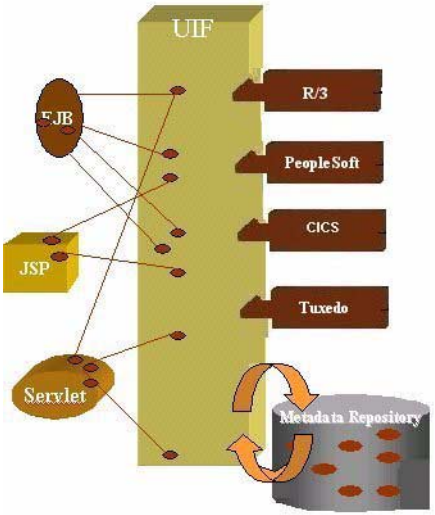
Figure 1-1 CICS Integration



The UIF enables development of server extensions that integrate with legacy CICS applications and systems, client-server applications, and third-party Internet solutions. These extensions provide a consistent access layer to disparate back-end systems, dramatically reducing development effort. The framework provides support for features such as object-pooling, distributed state and session management.

A generic data repository, also a part of the UIF, which is used to hold metadata parameters and other information about the back-end system. For example, the metadata often describes the physical connection between systems, the data that is available, and methods you can use to process data. See Figure 1-2.

Figure 1-2 The Unified Integration Framework



Since back-end systems are implemented differently, the details for each Enterprise Information System (EIS) implementation differ.

UIF Services

The UIF is a component of the iPlanet Application Server. The iPlanet Application Server plays a prominent role in a three-tier application model. See the Three-tier Application Model for a description. The UIF mediates between the iPlanet Application Server application and the EIS (back-end) tier, namely the data sources and databases.

The UIF provides an API to access the following services:

- Runtime
- Data Object Services
- Repository and Metadata Services

Runtime

The UIF runtime services supply core services for resource management, thread management, communication and life cycle management, and exception management. The UIF runtime services understand and interpret metadata repository contents.

Data Object Services

The Data Object Services implements universal data representation common to all connectors. See the *iPlanet Application Server Enterprise Connector for CICS Developer's Guide* for a description of data objects.

Repository and Metadata Services

The UIF repository and metadata services model a persistent information hierarchy that supports datatype definitions, and inheritance. It also manages the instances and reuse of data objects from datatype definitions.

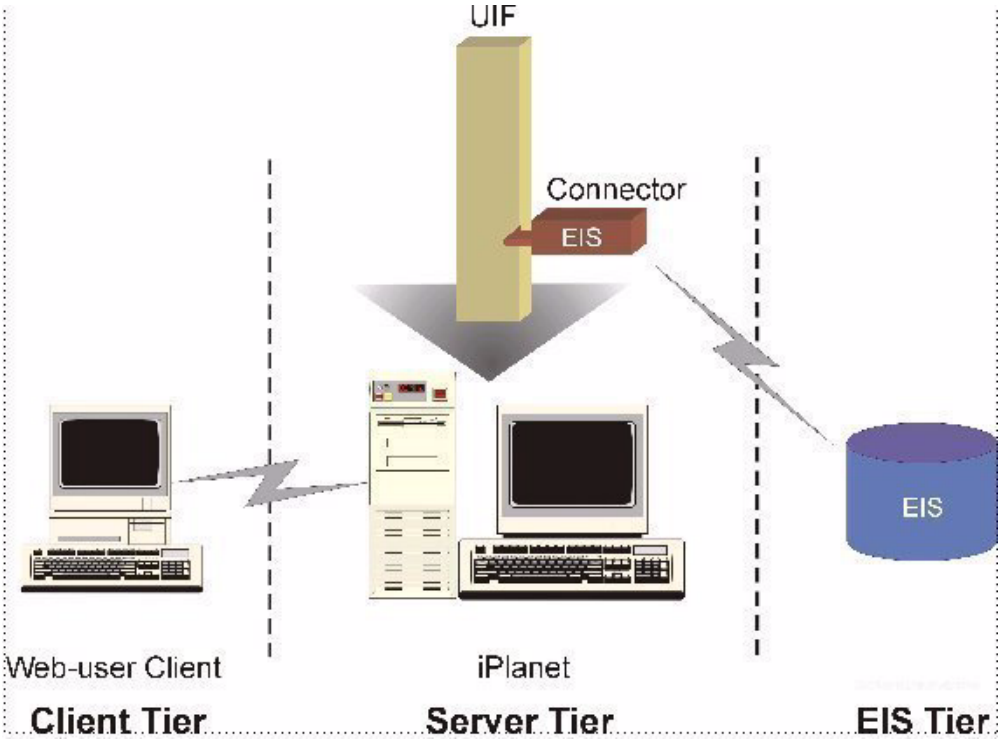
Three-tier Application Model

The machine and software involved are divided into the following three tiers:

- Client Tier
- Server Tier (iPlanet Application Server)
- EIS (Back-end) Tier

The connectors serve as an essential link allowing the server tier to communicate with the back-end tier, as shown in Figure 1-3. Communication between the application server and the back-end EIS is facilitated by the UIF API. This layer of functionality resides as an added layer to the iPlanet Application Server, and enables data communication with diverse back-end EIS's in a seamless and uniform manner.

Figure 1-3 Three-tier Web-based Computer Model



Client Tier

The client tier is represented as the user interface. Requests for data originate here, represented by web browsers or rich clients (such as a Java applet).

Server Tier

The server tier is represented by an application server, and optionally a web server such as the iPlanet Web Server Enterprise Edition. The server tier houses the business logic (your application servlets and/or Enterprise Java Beans), and provides scalability, high availability load balancing, and integration with a variety of data sources.

EIS (Back-end) Tier

The back-end tier is represented by Enterprise Resource Planning (ERP) systems or other back-end data systems such as CICS.

Enterprise Connector Tools for CICS

The Enterprise Connector Tools are as follows:

- Management Console - includes User Mapping and Data Mining Tools
  - User Mapping - allows you to map user IDs for access into the back-end system, and to edit and manage data sources.
  - Data Mining Tool - includes capabilities such as determining the available functions in the back-end system, translating and reformatting data, and loading data into the data repository.
- Repository Browser - allows you to browse data in the repository. You can view the available functions (input and output parameters) for the back-end system.

The Management Console

The iPlanet Application Server Enterprise Connector for CICS has an interactive Management Console Tool. The Management Console is used by the administrator to do the following:

- data mine function objects and data types from the back-end system metadata, and populate the repository with these data-mined definitions
- administer repository contents for creating new data sources, editing settings, and altering pooling settings
- populate user-mapping tables in the data repository

For more information on the Management Console, see Chapter 4.

The Repository Browser

The Repository Browser allows the user to explore the repository contents. While not used for editing, however, the Repository Browser provides import, export, and delete actions on repository nodes.

Repository contents are modified/populated using the *import* function. This feature is available as a browser function and as a command-line tool and imports XML under a specified repository node. Additionally, the export function is available as a browser function and as a command-line tool, and exports a selected subtree as an XML document. For more information on the Repository Browser, see Chapter 4.

National Language Support

The iPlanet Application Server Enterprise Connector for CICS provides full support for building applications that use the prevalent and popular mainframe character sets. You need to set both the client character set and the backend character set to work with other languages. See Table 4-2 for details on setting the character set.

Chapter 2

Installation

This chapter describes how to install, start, and uninstall the iPlanet Application Server Enterprise Connector for CICS.

This chapter contains information about the following topics:

- Software Prerequisites
- Hardware Requirements
- Installing the iAS Enterprise Connector for CICS on Windows NT/2000
- Uninstalling the iAS Enterprise Connector for CICS from Windows NT/2000
- Installing the iAS Enterprise Connector for CICS on Solaris
- Uninstalling the iAS Enterprise Connector for CICS from Solaris
- Upgrading or Re-installing Issues
- Post-installation Issues

Software Prerequisites

The following software prerequisites are necessary to install the iPlanet Application Server Enterprise Connector for CICS:

- iPlanet Application Server version 6.5
- iPlanet Application Server Unified Integration (UIF) version 6.5
- CICS software on the mainframe
- One of the following web browsers should be installed:
    - Netscape Communicator 4.5

- Microsoft Internet Explorer 4.0 (NT only)

- The iPlanet Application Server 6.5 must reside on the same system on which you are installing the connector.

Supported Versions of CICS

The *iPlanet Application Server Enterprise Connector for CICS* supports IBM CICS running on OS/390. The required releases are MVS 5.2.2 and CICS ESA 4.1 and above with current APARS.

iPlanet Application Server Enterprise Connector for CICS includes installation and configuration for both TCP/IP and SNA (APPC) protocols.

Hardware Requirements

The following hardware requirements are necessary for installing the connectors.

Computers and Operating Systems

- One of the following operating systems:
  - Microsoft Windows NT version 4.0 SP 6
  - Microsoft Windows2000
  - Sun SPARC running Solaris 2.6 or Solaris 8
- The MVS Operating System version 5.2.2 or later, or any version of OS390 must be installed before you can use the Listener:
- Communication protocol must be one of the following:
  - not running on TCP/IP. (use APPC)
  - IBM TCP/IP running on OS/390
  - CA TCPaccess running on OS/390

NOTE We work only with versions that are supported IBM or CA.

- Memory Per CPU: 256 MB minimum; 512 MB recommended

- Available disk space: 200 MB for NT/2000, 400 MB for Solaris

Preparing to Install the iAS Enterprise Connector for CICS

Before installing the iAS Enterprise connector for CICS complete the following steps:

- Verify the CICS Server connectivity.
- Stop the iPlanet Application Server if it is running.Without stopping the application server, the installation program may not be able to overwrite files currently in use.
- Read the *iAS Enterprise Connector for CICS Release Notes* for any last minute updates.

Installing the iAS Enterprise Connector for CICS on Windows NT/2000

The following describes the procedures for installing and uninstalling iPlanet Application Server for CICS on Windows NT/2000.

You must be logged in to Windows NT/2000 as the administrator to install the iAS Enterprise Connector for CICS.

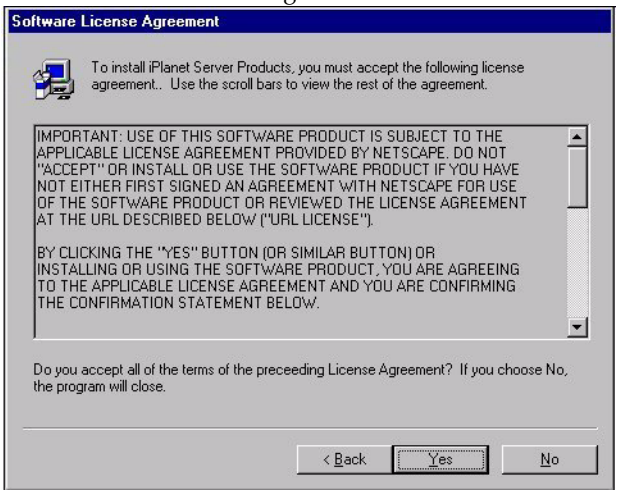
To Install the CICS Connector on Windows NT/2000

1. Read the README.txt file on the CD-ROM.
2. Copy `NT/inscics.zip` from the CD-ROM in to a temporary local directory.
3. Unzip `inscics.zip`, then run `setup.exe`.
4. Read the Welcome message and click on the Next button. See Figure 2-1.
5. Follow the instructions on the screen to progress through the introduction.

Figure 2-1 Welcome Message

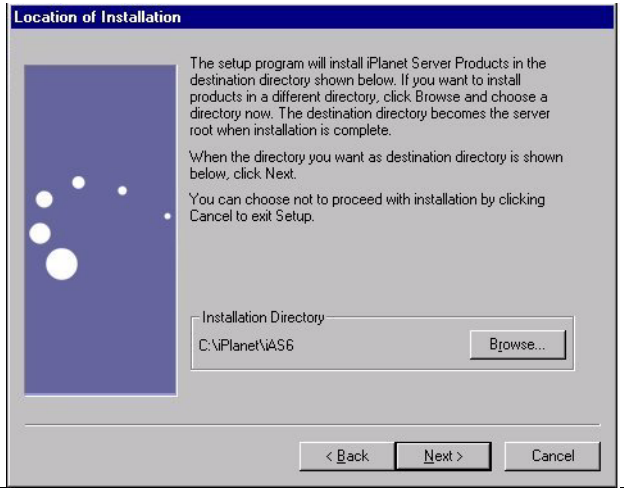
6. Thoroughly read and understand the licensing information before installing, and accept the license agreement. See Figure 2-2.

Figure 2-2 Software License Agreement



7. Choose your installation directory, then click Next. See Figure 2-3.

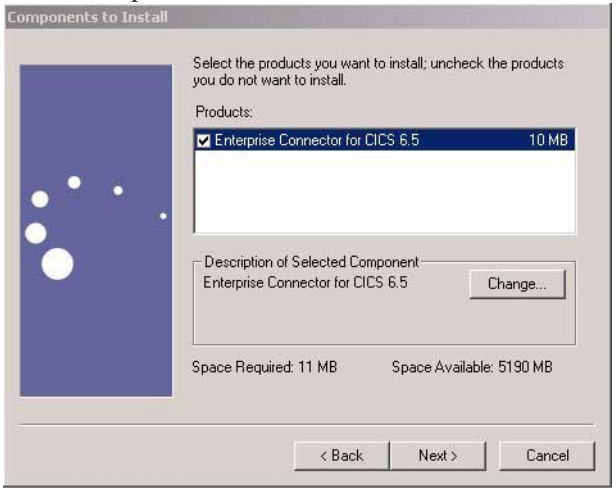
Figure 2-3 Location of Installation



NOTE You must install the iPlanet Application Server Enterprise Connector for CICS in the same directory as where you installed iPlanet Application Server and UIF.

8. Click Next to continue.

Figure 2-4 Components to Install

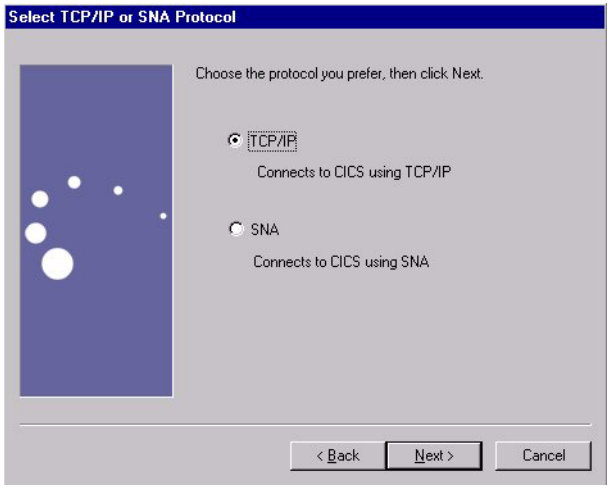


Select the Enterprise Connector for CICS 6.5 check box. See Figure 2-4.

9. Choose the protocol you prefer, either TCP/IP or SNA protocol, by selecting the appropriate button, and then click Next. See Figure 2-5.

NOTE See Chapter 3 "Installing the Mainframe Component" for a description of alternatives to connect the CICS connector to the mainframe.

Figure 2-5 Select TCP/IP or SNA Protocol

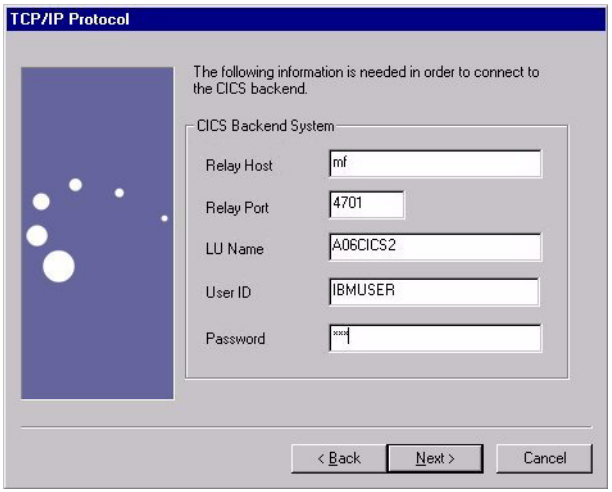


Continue with Step 10 if you selected TCP/IP protocol or continue with Step 11 if you selected the SNA protocol.

10. Type in your CICS Backend System definition for TCP/IP and click Next.

    The Configuration Summary shown in Figure 2-8 is displayed.

Figure 2-6 TCP/IP Protocol



11. Enter the CICS Backend System information for APPC and click Next.

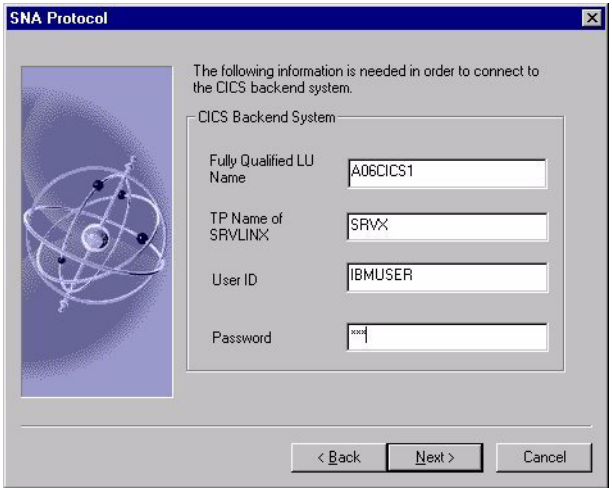
Table 2-1 describes the TCP/IP information

Table 2-1 TCP/IP Protocol CICS Backend System Information

Field	Definition
Relay Host	The address of the relay host that provides the service
Relay Port	The port number through which the relay intends to communicate.
LU Name	Name of LU of CICS (logon Applid)
User ID	User ID of the host.
Password	User password

The Configuration Summary shown in Figure 2-8 is displayed.

Figure 2-7 SNA Protocol



12. Type in the SNA protocol parameters into the text box.

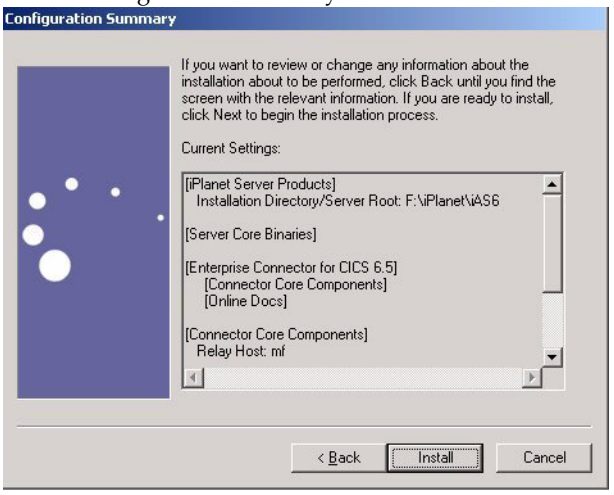
The parameters are defined in Table 2-2.

Table 2-2 SNA Protocol CICS Backend System Information

Field	Definition
LU Name	Name of LU of CICS (Logon Applid)
TP name of SRVLINK	Name of transaction of SRVLINK
User ID	User ID of the host
Password	User password.

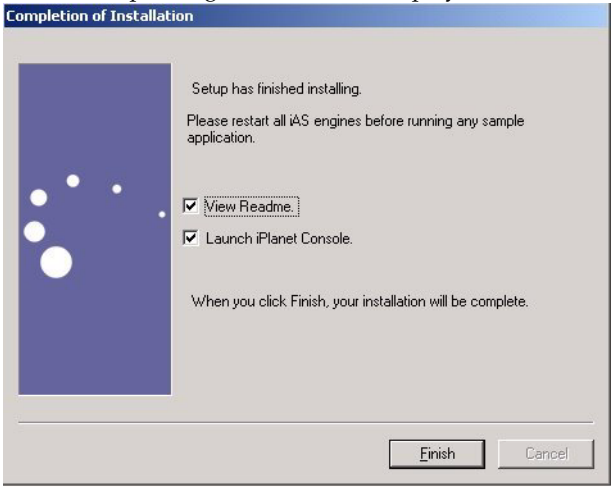
13. Press Next to display the configuration Summary screen.

Figure 2-8 Configuration Summary



14. Click Install if the information is correct.

Figure 2-9 Completiong of Installation Display



15. Click Finish and then proceed to install the ICU libraries.

ICU libraries

CAUTION **To complete the setup for the iPlanet Application Server Enterprise Connector for CICS, you must install the ICU libraries. Before proceeding to copy the ICU libraries, you must read and accept the ICU license agreement present in the ibm_icu_license.html file on the ICU CD.**

To install ICU:

- Uncompress the icu-NT.zip under the <iAS InstallDir>\ias\APPS\bin directory.

  This places the following directories under the <iAS InstallDir>\ias\APPS\bin directory:

  - icuuc.dll
  - icui18n.dll
  - icu (directory)

Post-installation

CAUTION You must restart your computer after you complete the installation.

After installing the Enterprise Connector for CICS on Windows NT/2000, proceed with the Mainframe component installation as described in Chapter 3.

If you selected a direct APPC connection from NT/2000, SNA/APPC on NT/2000 must be configured as explained in the following section.

SNA/APPC Configuration for Windows NT/2000

A sample configuration file for IBM Personal Communications (PCOMM) product for NT/2000 is enlisted, and supplied in:

```
<instdir>\APPS\adapters\CICS\backend\appconfig.ACG
```

Load this file into your PCOMM SNA Node Configuration as a basis for changes and adaptations to your SNA/APPC setup.

Code Example 2-1 PCOMM LU6.2 Configuration Definitions on a PC

```
*TSTue Mar  7 10:57:58 2000
NODE=(
     ANYNET_SUPPORT=NONE
     CP_ALIAS=SHMUELNT
     DEFAULT_PREFERENCE=NATIVE
     DISCOVERY_SUPPORT=NO
     FQ_CP_NAME=P390.SHMUELNT
     NODE_ID=05D00000
     NODE_TYPE=END_NODE
     REGISTER_WITH_CDS=1
     REGISTER_WITH_NN=1
)
PORT=(
     PORT_NAME=LANX_04
     ACTIVATION_DELAY_TIMER=30
     DELAY_APPLICATION_RETRIES=1
     DLC_DATA=00000000000004
     DLC_NAME=LAN
     IMPLICIT_CP_CP_SESS_SUPPORT=1
     IMPLICIT_DEACT_TIMER=0
     IMPLICIT_DSPU_SERVICES=NONE
     IMPLICIT_HPR_SUPPORT=1
     IMPLICIT_LIMITED_RESOURCE=NO
     IMPLICIT_LINK_LVL_ERROR=0
     LINK_STATION_ROLE=NEGOTIABLE
     MAX_ACTIVATION_ATTEMPTS=10
     MAX_IFRM_RCVD=8
     MAX_RCV_BTU_SIZE=32767
     PORT_TYPE=SATF
     RETRY_LINK_ON_DISCONNECT=1
     RETRY_LINK_ON_FAILED_START=1
     RETRY_LINK_ON_FAILURE=1
     PORT_LAN_SPECIFIC_DATA=(
          ACK_DELAY=100
          ACK_TIMEOUT=3000
          ADAPTER_NUMBER=9999
          BUSY_STATE_TIMEOUT=60
          IDLE_STATE_TIMEOUT=30
          LOCAL_SAP=04
          MAX_RETRY=10
          OUTSTANDING_TRANSMITS=16
          POLL_TIMEOUT=3000
          POOL_SIZE=16
          REJECT_RESPONSE_TIMEOUT=30
          TEST_RETRY_INTERVAL=8
          TEST_RETRY_LIMIT=5
          XID_RETRY_INTERVAL=8
          XID_RETRY_LIMIT=5
     )
```

Code Example 2-1 PCOMM LU6.2 Configuration Definitions on a PC *(Continued)*

```
)
LINK_STATION=(
     LS_NAME=LINK0000
     ACTIVATE_AT_STARTUP=1
     ACTIVATION_DELAY_TIMER=-1
     ADJACENT_NODE_TYPE=LEARN
     AUTO_ACTIVATE_SUPPORT=0
     CP_CP_SESS_SUPPORT=1
     DEFAULT_NN_SERVER=0
     DELAY_APPLICATION_RETRIES=0
     DEST_ADDRESS=42000000006604
     DISABLE_REMOTE_ACT=0
     DSPU_SERVICES=NONE
     ETHERNET_FORMAT=1
     HPR_LINK_LVL_ERROR=0
     HPR_SUPPORT=0
     INHERIT_PORT_RETRY_PARMS=1
     LIMITED_RESOURCE=NO
     LINK_DEACT_TIMER=0
     LINK_STATION_ROLE=USE_ADAPTER_DEFAULTS
     MAX_ACTIVATION_ATTEMPTS=-1
     MAX_IFRM_RCVD=0
     MAX_SEND_BTU_SIZE=32767
     NODE_ID=05D00000
     PORT_NAME=LANX_04
     RETRY_LINK_ON_DISCONNECT=0
     RETRY_LINK_ON_FAILED_START=0
     RETRY_LINK_ON_FAILURE=0
     SOLICIT_SSCP_SESSION=0
     TG_NUMBER=0
     USE_DEFAULT_TG_CHARS=1
)
DLUR_DEFAULTS=(
     DEFAULT_PU_NAME=SHMUELNT
     DLUS_RETRY_LIMIT=65535
     DLUS_RETRY_TIMEOUT=5
)
MODE=(
     MODE_NAME=BLANK
     AUTO_ACT=0
     COMPRESSION=PROHIBITED
     COS_NAME=#CONNECT
     DEFAULT_RU_SIZE=1
     MAX_NEGOTIABLE_SESSION_LIMIT=256
     MAX_RU_SIZE_UPPER_BOUND=1024
     MIN_CONWINNERS_SOURCE=128
     PLU_MODE_SESSION_LIMIT=256
     RECEIVE_PACING_WINDOW=3
)
MODE=(
     MODE_NAME=#BATCH
     AUTO_ACT=0
     COMPRESSION=PROHIBITED
     COS_NAME=#BATCH
```

Code Example 2-1 PCOMM LU6.2 Configuration Definitions on a PC *(Continued)*

```
      DEFAULT_RU_SIZE=0
      MAX_NEGOTIABLE_SESSION_LIMIT=256
      MAX_RU_SIZE_UPPER_BOUND=2048
      MIN_CONWINNERS_SOURCE=128
      PLU_MODE_SESSION_LIMIT=256
      RECEIVE_PACING_WINDOW=20
)
MODE=(
      MODE_NAME=#BATCHSC
      AUTO_ACT=0
      COMPRESSION=PROHIBITED
      COS_NAME=#BATCHSC
      DEFAULT_RU_SIZE=1
      MAX_NEGOTIABLE_SESSION_LIMIT=256
      MAX_RU_SIZE_UPPER_BOUND=2048
      MIN_CONWINNERS_SOURCE=128
      PLU_MODE_SESSION_LIMIT=256
      RECEIVE_PACING_WINDOW=3
)
MODE=(
      MODE_NAME=#INTER
      AUTO_ACT=0
      COMPRESSION=PROHIBITED
      COS_NAME=#INTER
      DEFAULT_RU_SIZE=1
      MAX_NEGOTIABLE_SESSION_LIMIT=256
      MAX_RU_SIZE_UPPER_BOUND=4096
      MIN_CONWINNERS_SOURCE=128
      PLU_MODE_SESSION_LIMIT=256
      RECEIVE_PACING_WINDOW=20
)
MODE=(
      MODE_NAME=#INTERSC
      AUTO_ACT=0
      COMPRESSION=PROHIBITED
      COS_NAME=#INTERSC
      DEFAULT_RU_SIZE=1
      MAX_NEGOTIABLE_SESSION_LIMIT=256
      MAX_RU_SIZE_UPPER_BOUND=2048
      MIN_CONWINNERS_SOURCE=128
      PLU_MODE_SESSION_LIMIT=256
      RECEIVE_PACING_WINDOW=7
)
MODE=(
      MODE_NAME=APPCMODE
      AUTO_ACT=0
      COMPRESSION=PROHIBITED
      COS_NAME=#CONNECT
      DEFAULT_RU_SIZE=1
      MAX_NEGOTIABLE_SESSION_LIMIT=128
      MAX_RU_SIZE_UPPER_BOUND=4096
      MIN_CONWINNERS_SOURCE=16
      PLU_MODE_SESSION_LIMIT=32
      RECEIVE_PACING_WINDOW=1
```

Code Example 2-1 PCOMM LU6.2 Configuration Definitions on a PC *(Continued)*

```
)
MODE=(
     MODE_NAME=QPCSUPP
     AUTO_ACT=0
     COMPRESSION=PROHIBITED
     COS_NAME=#CONNECT
     DEFAULT_RU_SIZE=1
     MAX_NEGOTIABLE_SESSION_LIMIT=52
     MAX_RU_SIZE_UPPER_BOUND=1024
     MIN_CONWINNERS_SOURCE=26
     PLU_MODE_SESSION_LIMIT=52
     RECEIVE_PACING_WINDOW=2
)
MODE=(
     MODE_NAME=QSERVER
     AUTO_ACT=0
     COMPRESSION=PROHIBITED
     COS_NAME=#CONNECT
     DEFAULT_RU_SIZE=1
     MAX_NEGOTIABLE_SESSION_LIMIT=64
     MAX_RU_SIZE_UPPER_BOUND=1024
     MIN_CONWINNERS_SOURCE=0
     PLU_MODE_SESSION_LIMIT=64
     RECEIVE_PACING_WINDOW=7
)
MODE=(
     MODE_NAME=SNASVCMG
     AUTO_ACT=0
     COMPRESSION=PROHIBITED
     COS_NAME=SNASVCMG
     DEFAULT_RU_SIZE=0
     MAX_NEGOTIABLE_SESSION_LIMIT=2
     MAX_RU_SIZE_UPPER_BOUND=512
     MIN_CONWINNERS_SOURCE=1
     PLU_MODE_SESSION_LIMIT=2
     RECEIVE_PACING_WINDOW=1
)
PARTNER_LU=(
     FQ_PLU_NAME=P390.A06CICS2
     ADJACENT_CP_NAME=P390.P390SSCP
     CONV_SECURITY_VERIFICATION=1
     MAX_MC_LL_SEND_SIZE=32767
     PARALLEL_SESSION_SUPPORT=1
     PARTNER_LU_ALIAS=A06CICS2
     PREFERENCE=USE_DEFAULT_PREFERENCE
)
CPIC_SIDE_INFO=(
     SYM_DEST_NAME=SRVX
     CONVERSATION_SECURITY_TYPE=NONE
     MODE_NAME=APPCMODE
     PARTNER_LU_NAME=P390.A06CICS2
     TP_NAME=SRVX
     TP_NAME_TYPE=APPLICATION_TP
)
```

Code Example 2-1 PCOMM LU6.2 Configuration Definitions on a PC *(Continued)*

```
ADJACENT_NODE=(
     FQ_CP_NAME=P390.P390SSCP
     LU_ENTRY=(
          FQ_LU_NAME=P390.A06CICS2
     )
)
SHARED_FOLDERS=(
     CACHE_SIZE=256
)
VERIFY=(
     CFG_MODIFICATION_LEVEL=13
     CFG_VERSION_LEVEL=1
)
```

To Initialize the SNA/APPC Protocol Stack in Windows NT/2000

1. Select Start > Programs > IBM Personal Communications > Administrative and PD Aids > SNA Node Operations.
2. Select Operations > Start Node.

   The Personal Communications Node Operations window appears.
3. Select Operations > CNOS Initialize.

Uninstalling the iAS Enterprise Connector for CICS from Windows NT/2000

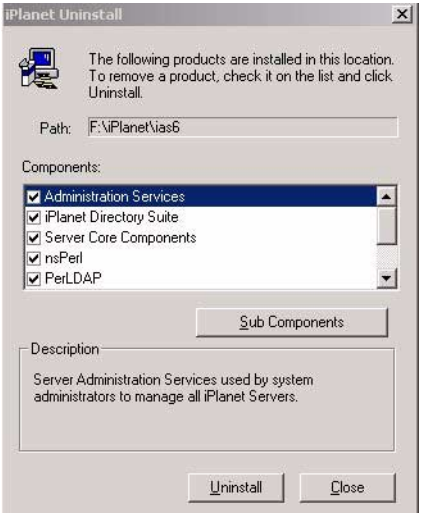
This procedure is used to uninstall *iPlanet Application Server Enterprise Connector for CICS from Windows NT/2000*.

To Uninstall the Enterprise Connector for CICS from Windows NT/2000

1. Select Start > Programs - iPlanet Application Server 6.5 > Uninstaller.

   The iPlanet Uninstall window is displayed. See Figure 2-10.

Figure 2-10 iPlanet Uninstaller Window



2. Deselect all components for iPlanet Application Server *except* iPlanet Application Server Enterprise Connector for CICS.

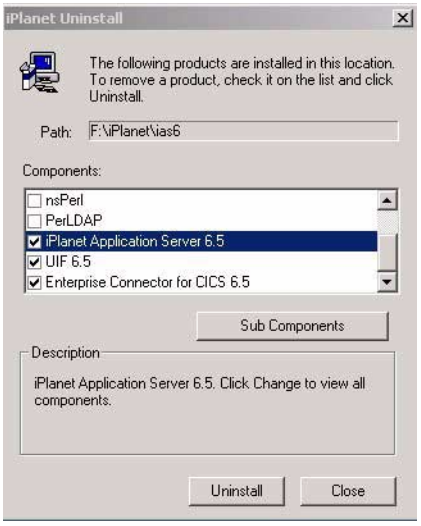
   The components must be deselected in the following sequence.

CAUTION The uninstaller checks for dependencies between subcomponents and the main component. If you try to deselect a component and its subcomponent are toggled on, a warning message appears.

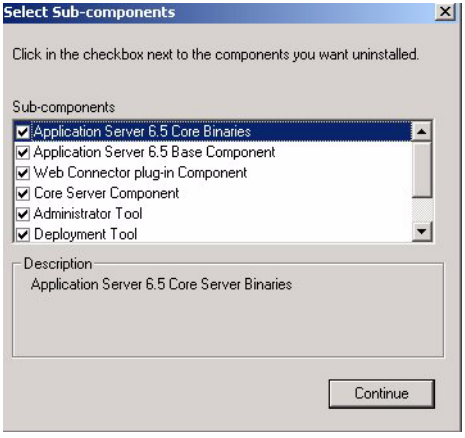
   a. Server Core Components
   b. Administration Services
   c. iPlanet Directory Suite
   d. nsPerl
   e. PerLDAP

3. Select iPlanet Application Server 6.5 as shown in Figure 2-11 and then press Sub Components.

Figure 2-11 iPlanet Uninstaller Window (Bottom)



4. Deselect iPlanet Application Server 6.5 Sub Components.



The components of the iPlanet Application Server 6.5 must be deselected in the following sequence.

I. Application Server 6.5 Core Binaries

II. Application Server 6.5 Base Component

III. Web Connector plug-in Component

IV. Administrator Tool

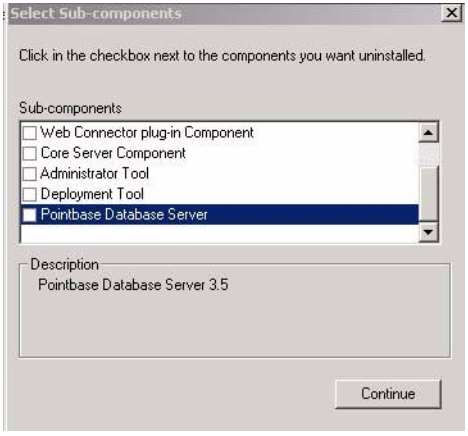
V. Core Server Component

VI. Deployment Tool

VII. Pointbase Database Server

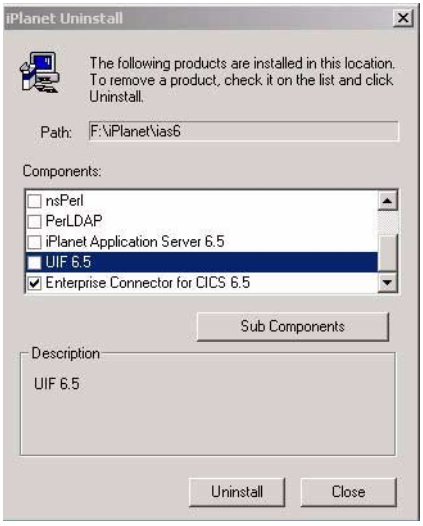
When you have finished deselecting the check boxes, the Select Sub-components dialog box should be as shown in Figure 2-12.

Figure 2-12 Deselected iPlanet Application Server 6.5 Components



5. Click Continue when you have finished deselecting the sub-components to redisplay the main uninstall menu.

Figure 2-13 Main uninstall



6. Deselect UIF 6.5
7. Click Uninstall.

   The connector is uninstall without any additional messages.

Installing the iAS Enterprise Connector for CICS on Solaris

The following is the installation process for the *iPlanet Application Server Enterprise Connector for CICS* on a Solaris operating system.

To ensure that you have everything necessary to complete the installation, review this section before starting the installation.

Pre-installation

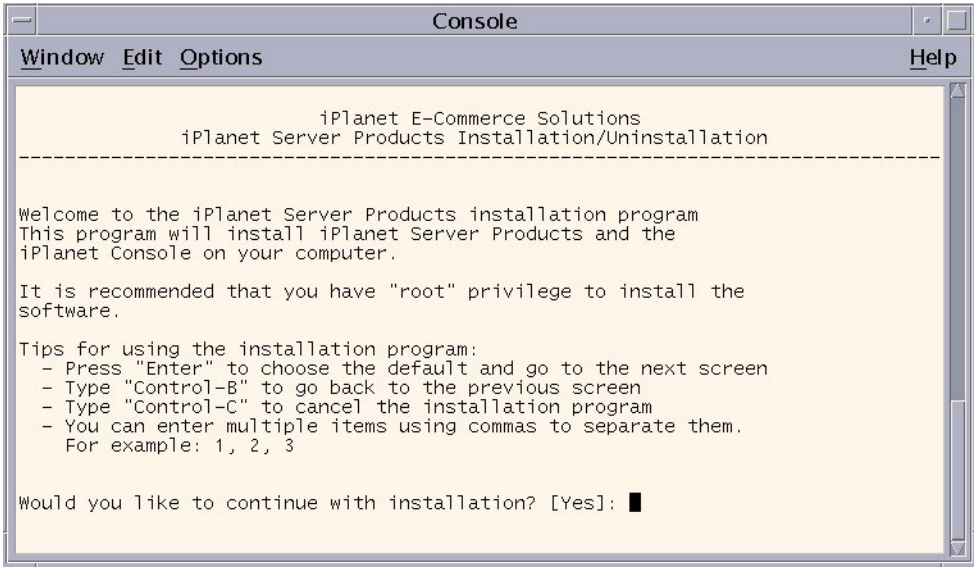
1. Insert the iAS Enterprise Connector for CICS Install CD-ROM into the CD-ROM drive.
2. Mount the CD-ROM, for example, `/cdrom/cdrom0`.
3. Read the README.txt file on the CD-ROM.

To Install the CICS Connector on Solaris

1. Copy the `Solaris/inscics.tar` file from the CD-ROM into a temporary local directory.
2. Untar the `inscics.tar` file.

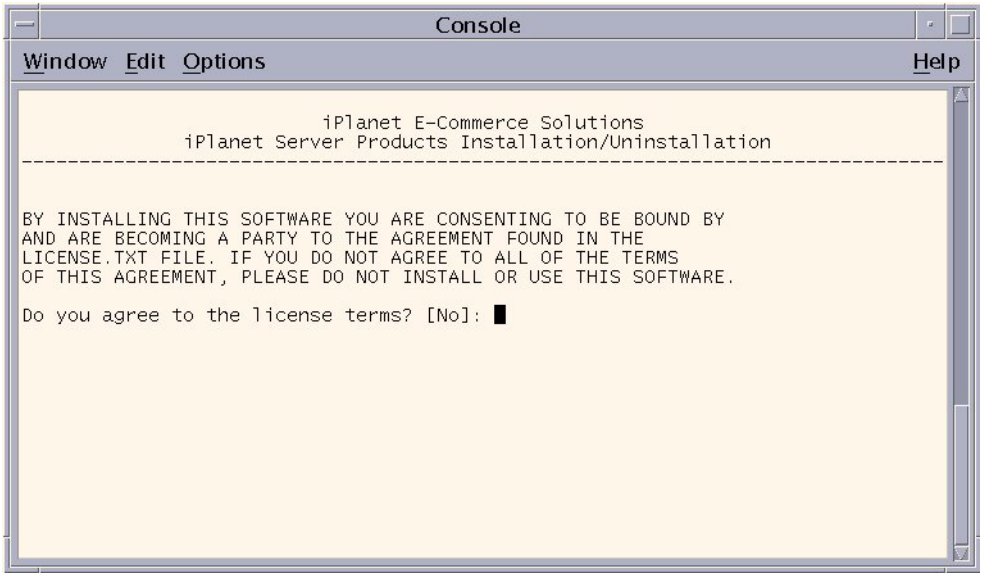
   For example: `tar -xvf inscics.tar`
3. Run the `./setup` command from the `inscics` directory
4. Figure 2-14 is displayed.

Figure 2-14 Welcome



5. Type Yes to continue. The License screen appears. See Figure 2-15.

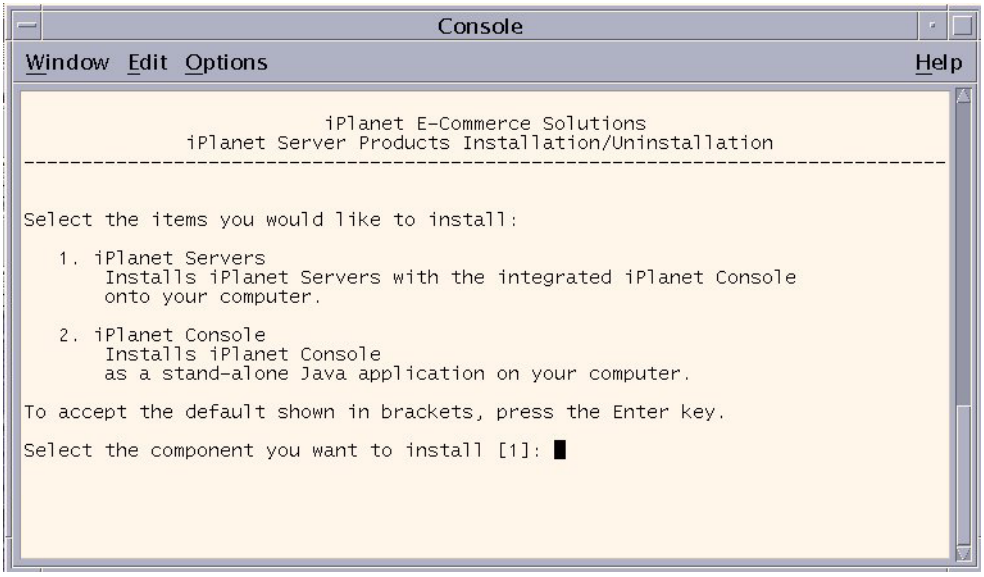
Figure 2-15 License



6. Thoroughly read and understand the licensing information before installing and then type Yes if you agree.

   The Selection message appears. See Figure 2-16.

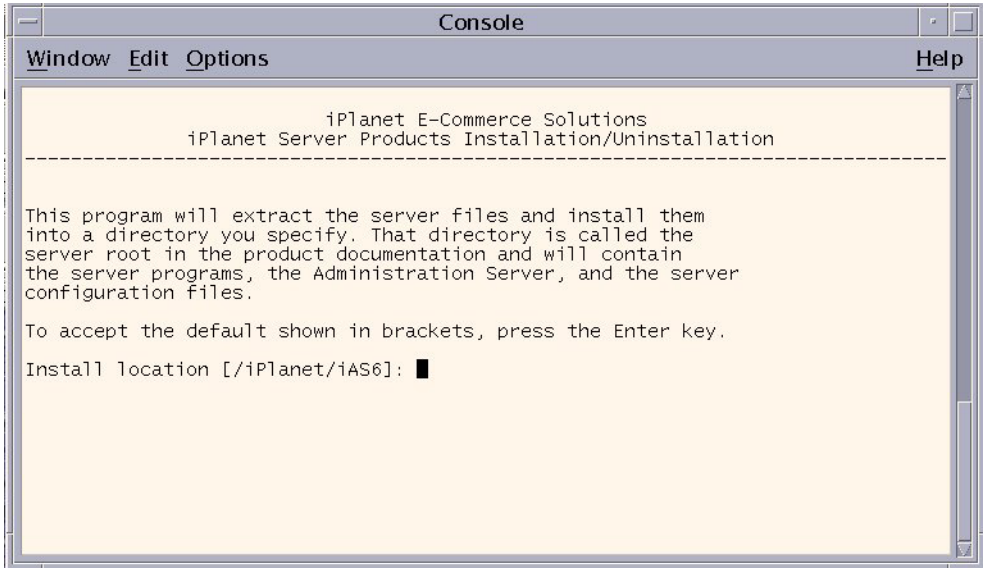
Figure 2-16 Selection



7. Press Enter to select the iPlanent Servers..

   The Location message appears. See Figure 2-17.

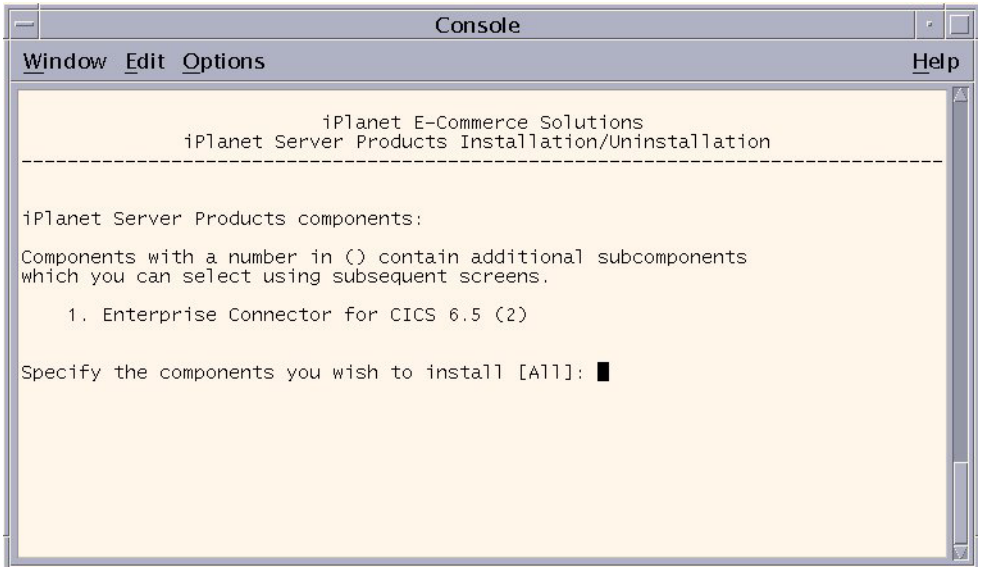
Figure 2-17 Location



Type in the location where you installed the iPlanet Application Server and then press Enter.

8. The Components message appears. See Figure 2-18.

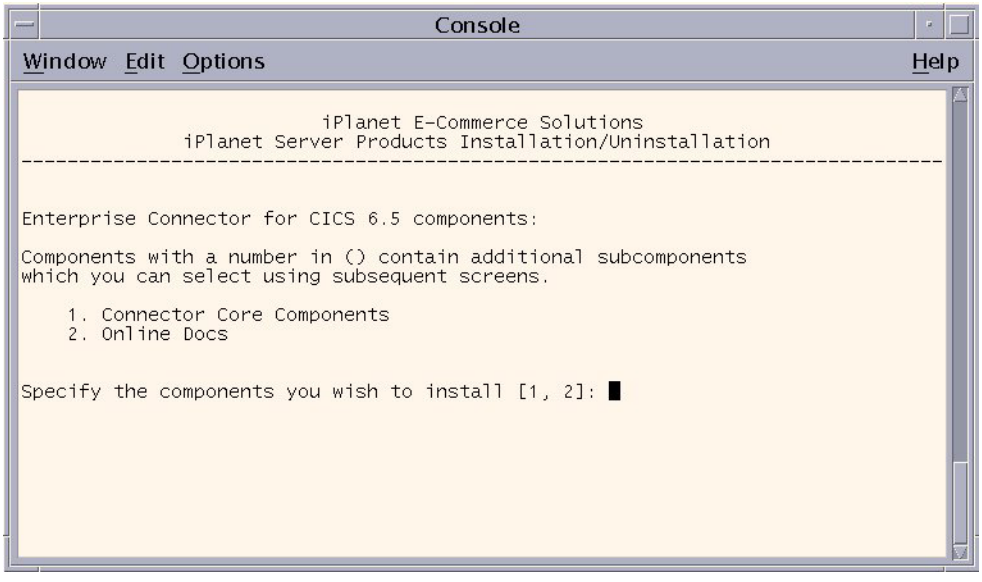
Figure 2-18 Components



9. Press Enter.

The Subcomponents message appears. See Figure 2-19.

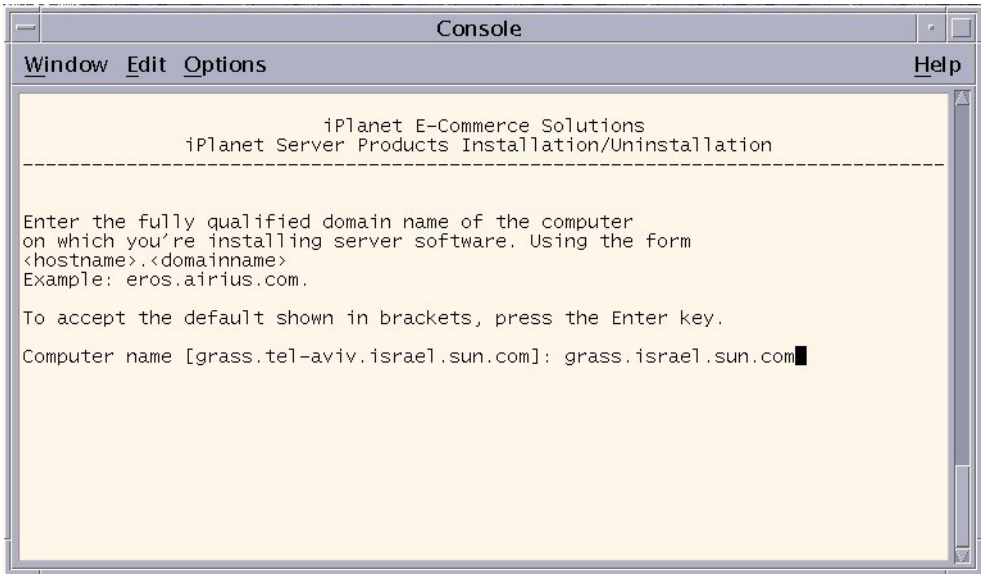
Figure 2-19 Subcomponents



10. Press Enter.

The Domain Name message appears. See Figure 2-20.

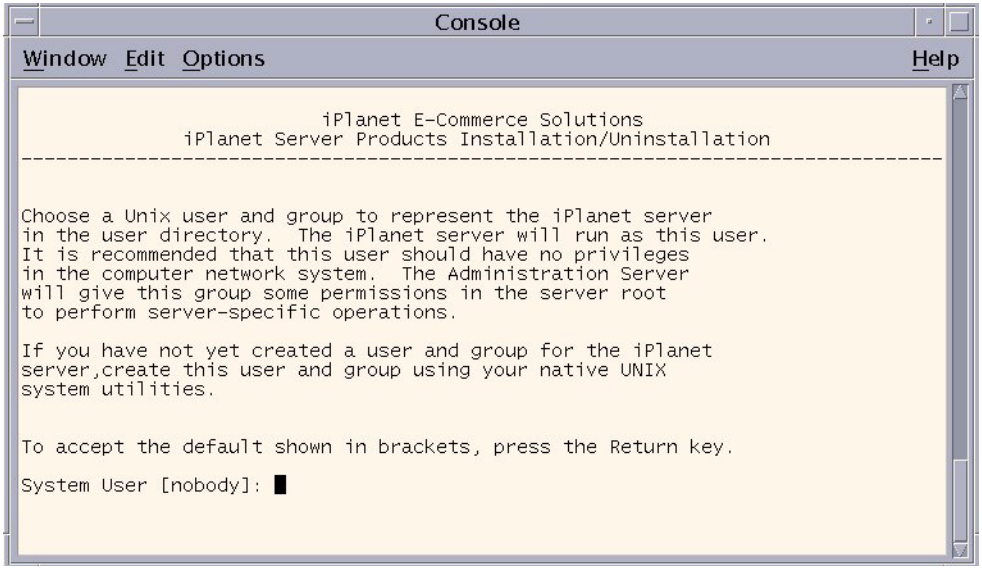
Figure 2-20 Domain Name



11. Click Enter to accept the default or enter another name `<host.domain>`.

The System User message appears. See Figure 2-21.

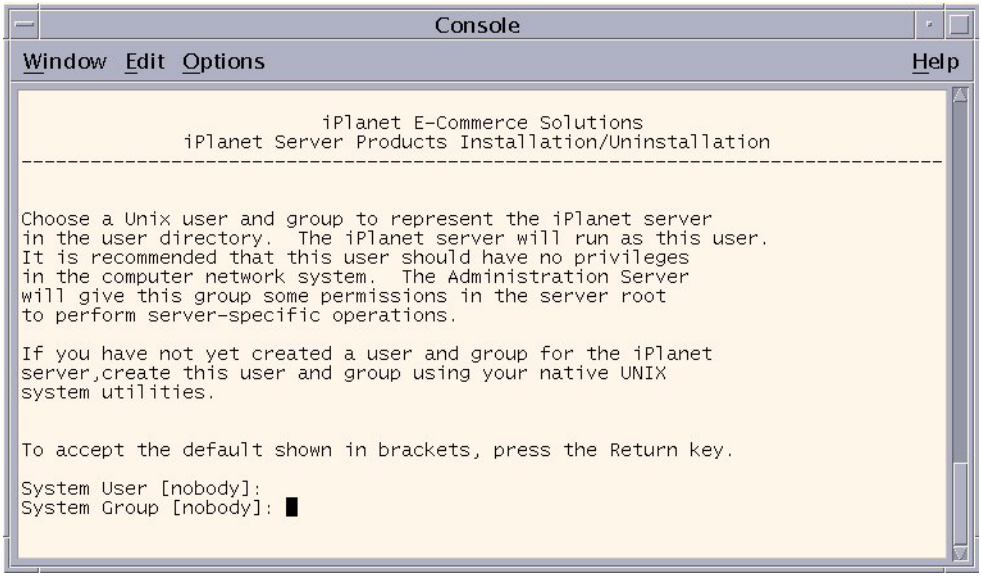
Figure 2-21 System User



12. Enter in a System User, or accept the default shown in brackets, and press Enter.

    The System Group message appears. See Figure 2-22.

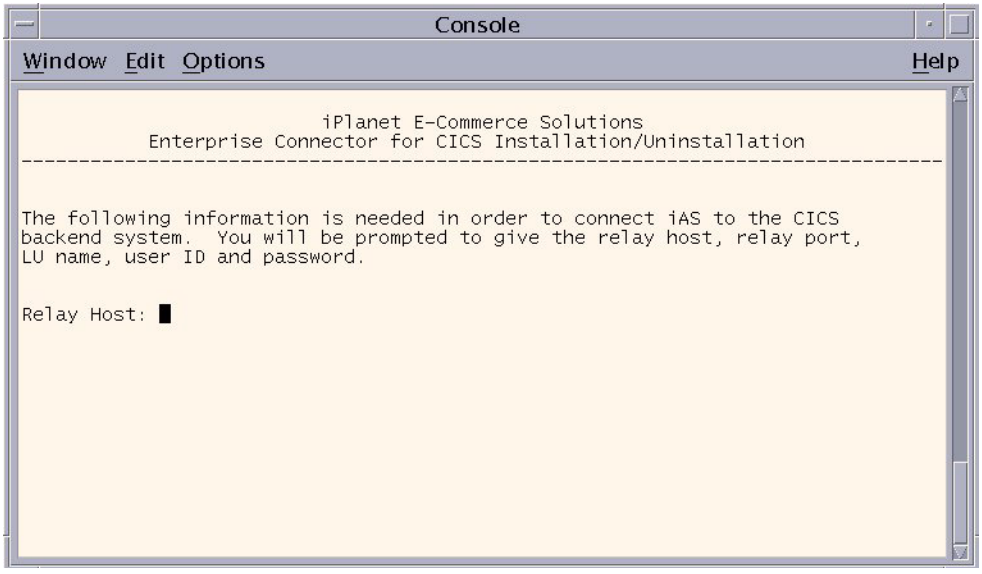
Figure 2-22 System Group



13. Enter in a System Group, or accept the default shown in brackets, and press Enter.

    The Relay Host message appears. See Figure 2-23.

Figure 2-23 Relay Host

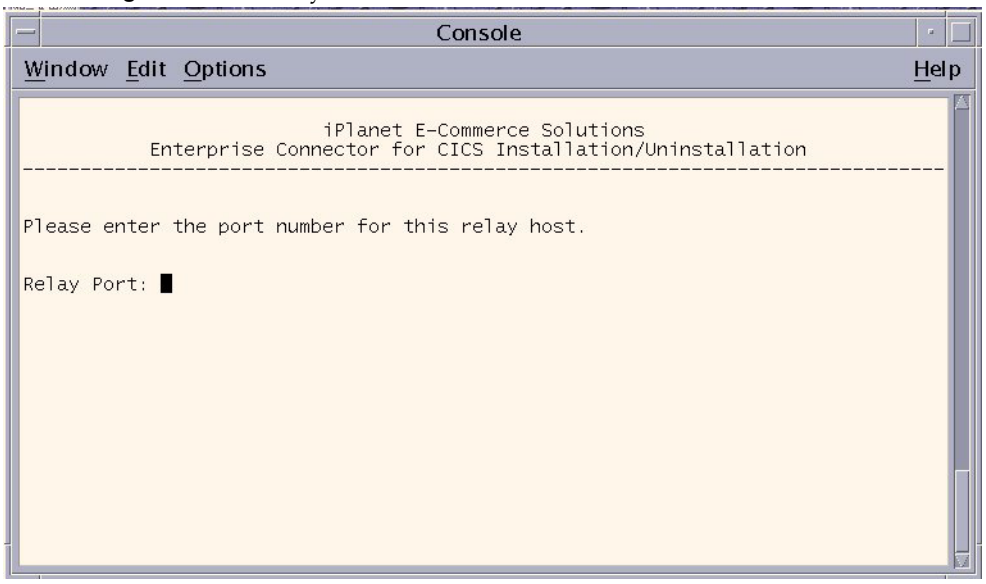


NOTE The following steps are concerned with information from the mainframe.

14. Type in the Mainframe TCP/IP Host name and press Enter.

The Relay Port message appears. See Figure 2-24.

Figure 2-24 Relay Port

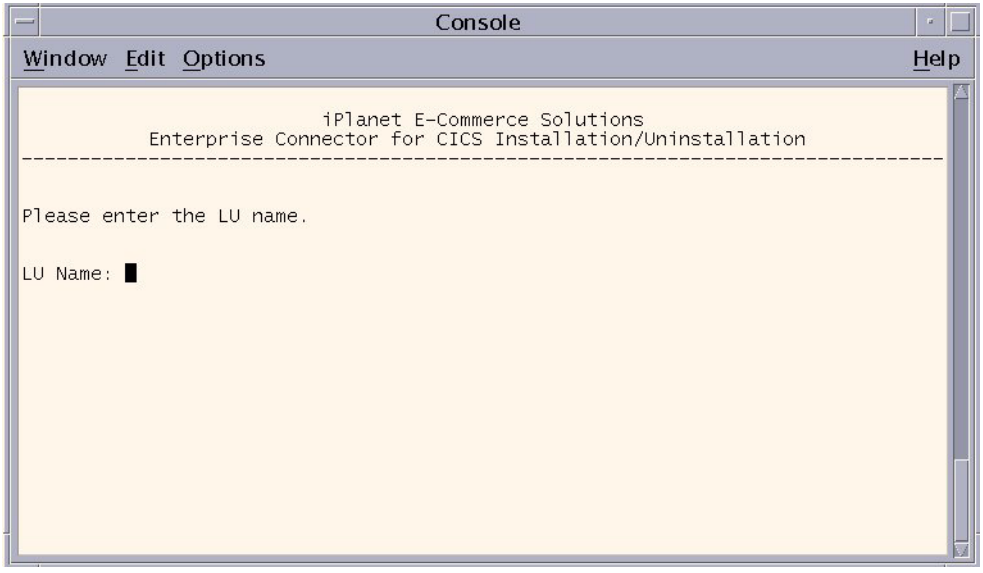


15. Type in the Relay Port, and press Enter.

NOTE The Relay Port number is the same as the Port number in Table 3-8.

The LU (logical unit) message appears. See Figure 2-25.

Figure 2-25 Logical Unit

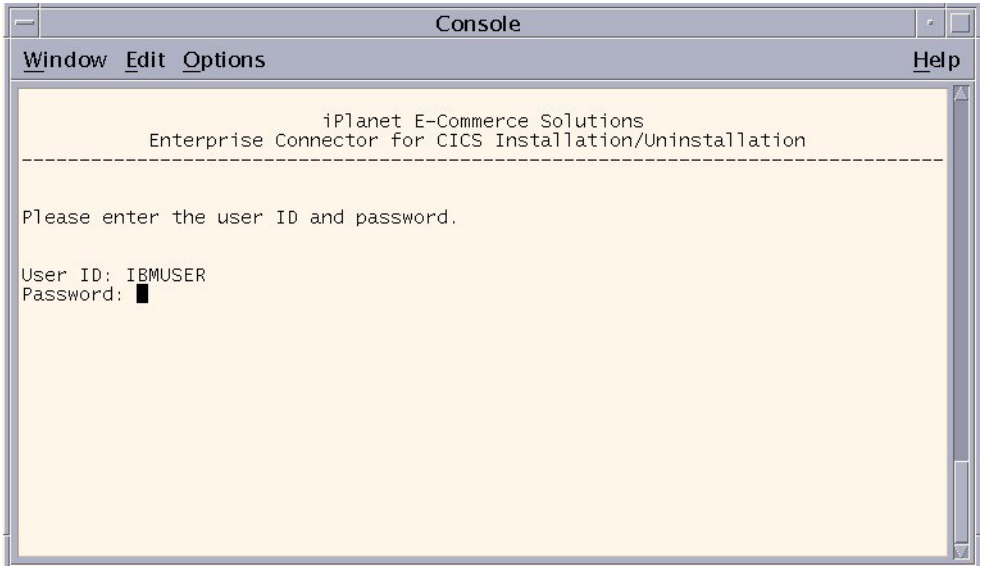


16. Enter the LU name and press Enter.

    The LU name is the VTAM application LU name for CICS. The LU name must be upper case.

    The User ID and Password screen appears. See Figure 2-26.

Figure 2-26 User ID and Password

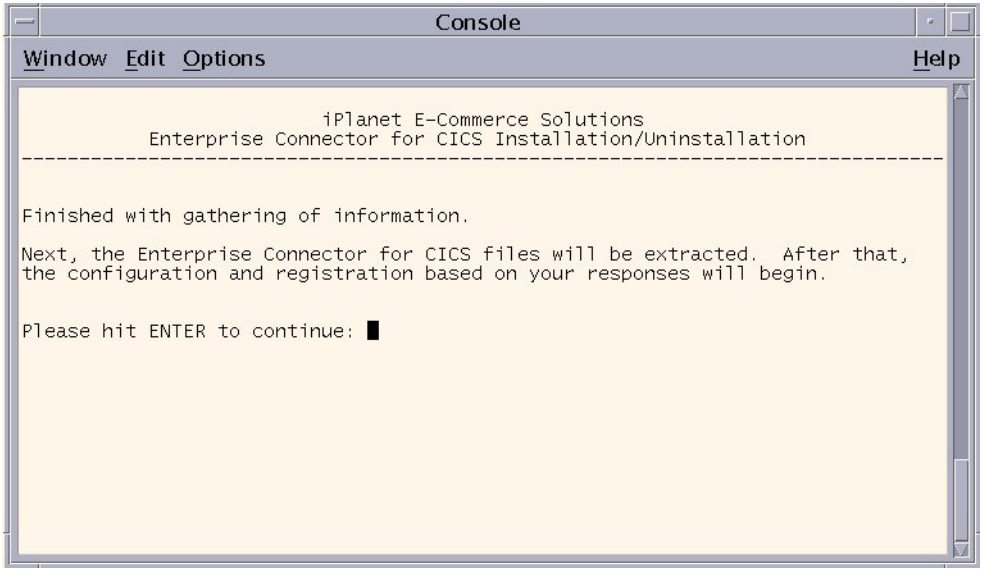


17. Type in the User ID and Password, and press Enter.

    This must be a valid User ID and Password pair as defined in the CICS backend and both names must be upper case.

    The final installation screen appears. See Figure 2-27.

Figure 2-27 Final Installation Screen

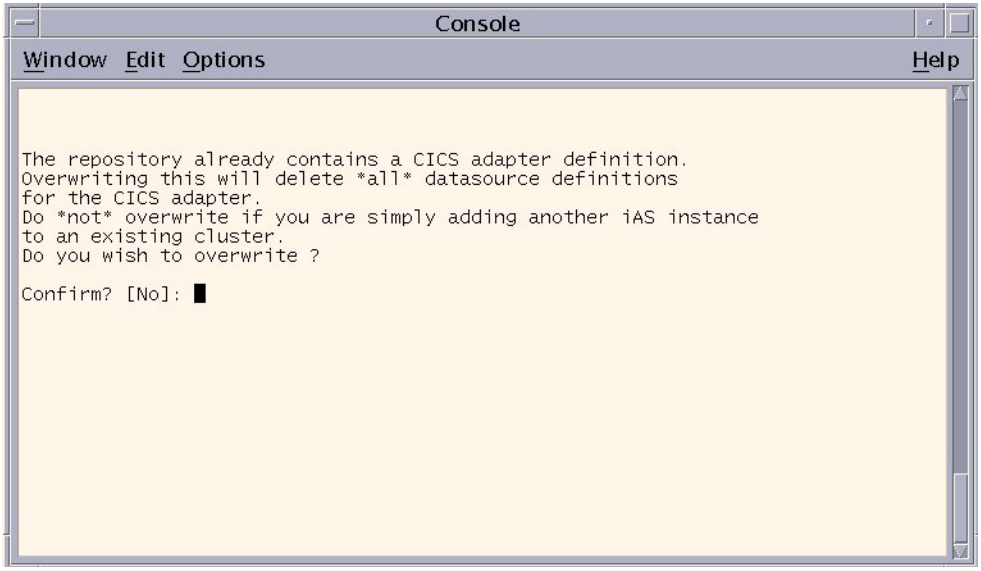


18. Press Enter to continue.

    A Work-in-Progress screen will appear as the connector and related files are installed. When finished, you will be instructed to restart all iAS engines before running any sample application.

19. The following screen will be displayed if you already have an iAS instance installed.

Figure 2-28 Addition iAS Instance to and Existing Cluster



Type either Yes or No depending upon whether you want to create an additional instance.

ICU Libraries

To complete the setup for the iPlanet Application Server Enterprise Connector for CICS, you must install the ICU libraries. Before proceeding to copy the ICU libraries, you must read and accept the ICU license agreement present in the ibm_icu_license.html file on the ICU CD.

To install the ICU libraries:

Uncompress the icu-SOLARIS.tar under the <iAS installDir>/ias/APPS/bin directory.

This places the following files/directories under the <iAS_installDir>/ias/APPS/bin directory:

- libicu-i18n.so
- libicu-cu.so
- icu (directory)

Post-installation

You must restart iAS services.

Continue with Chapter 3, “Installing the Mainframe Component”.

Uninstalling the iAS Enterprise Connector for CICS from Solaris

This procedure is used to uninstall *iPlanet Application Server Enterprise Connector for CICS* from Solaris.

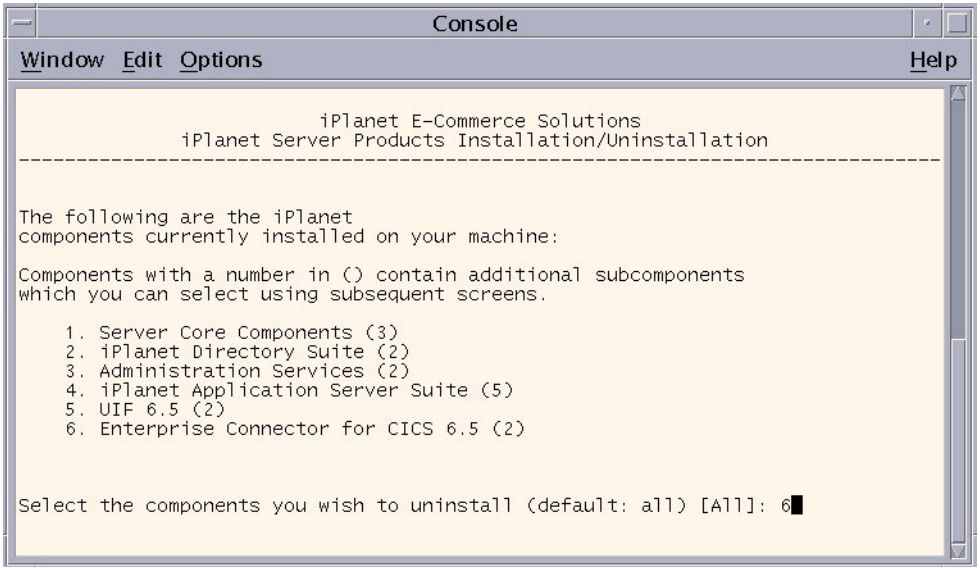
To Uninstall the CICS Connector from Solaris

1. Go to the installation directory */<ias_inst_dir>* and enter the following command:

   ```
   ./uninstall
   ```

   A list of uninstall component options appears. See Figure 2-29.

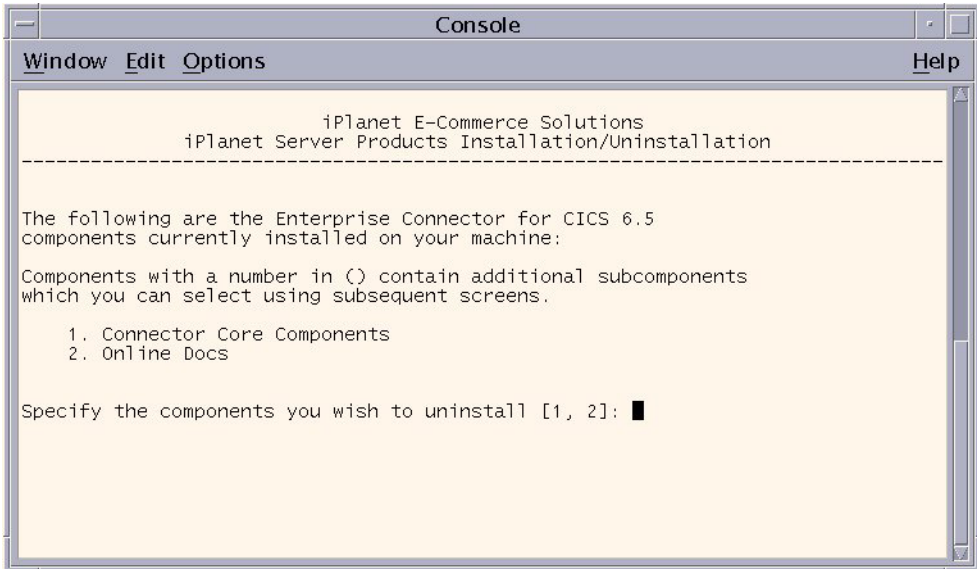
Figure 2-29 Uninstall Component Options



2. Select the specific product components you want to uninstall and click Enter.

   The Uninstall Subcomponents message appears. See Figure 2-30.

Figure 2-30 Uninstall Subcomponents



3. Press Enter.

   The Uninstalling message appears.

   The Enterprise Connector for CICS 6.5 has been uninstalled. Some files may remain. Check for these files and manually remove them.

Upgrading or Re-installing Issues

If upgrading an earlier version or reinstalling the current version of the iAS Enterprise Connector for CICS, the installation procedure overwrites the currently installed files. Uninstalling the current version is not required.

Multi-connector Installation

The Uninstaller removes common files in multi-connector installation.When uninstalling one of several connectors, such as R/3, PeopleSoft, or CICS, the common directories /ias/APPS/console and /ias/APPS/ecu may be deleted. You must copy these directories back to APPS directory from /ias/APPS/UIFbackup for the CICS connector to function correctly. Installation of multiple enterprise connectors for multiple EISs (R/3, Tuxedo, and PeopleSoft) on the same iPlanet Application Server is allowed.

NOTE To avoid problems with overwriting files in use, stop the iPlanet Application Server before installing any new software. Always backup the current files prior to installation including the UIF Repository content.

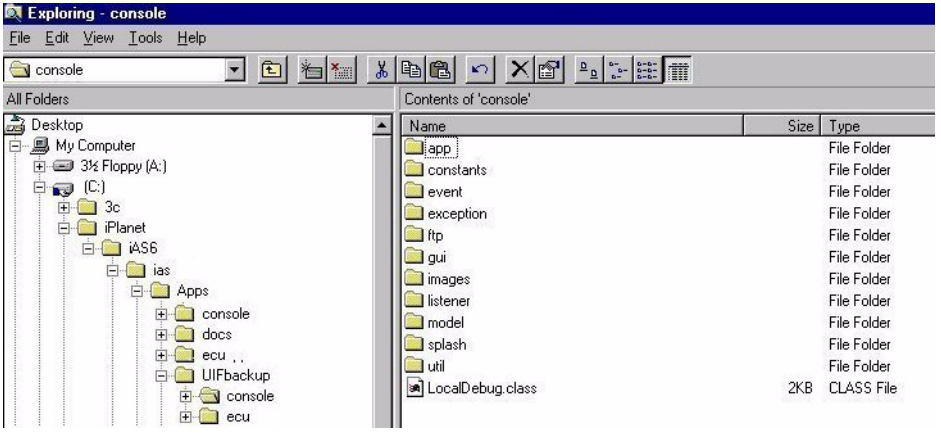
To Uninstall a Connector on Windows NT/2000

If you want to uninstall a connector from a Windows NT/2000 machine that has more than one connector installed, perform the following tasks.

1. Uninstall the connector. Refer to Chapter 2 in the Administrator's Guide.

   During the uninstallation procedure the connector files shown in the right pane, see Figure 2-31, are removed from the iPlanet/iAS6/ias directory and moved to the UIFbackup/console directory. The connector files need to be copied back to their original location.

Figure 2-31 Uninstalling Connector



2. Copy the console specific directories, the top level directory and the connector specific directories to the ias/Apps/console for the following folders:
   - constants
   - exceptions
   - ftp
   - gui
   - model
3. All the other directories, with all their included files, and the LocalDebug.class should be copied back.
4. Copy the following connector files to the console directory:
   - Appltab
   - ldobuffer.dll.

To Uninstall a Connector on Solaris

If you want to uninstall a connector from a Solaris machine that has more than one connector installed, perform the following tasks.

1. Uninstall the connector. Refer to Chapter 2 in the Administrator's Guide.

   During the uninstallation procedure the connector files are removed from the iPlanet/iAS6/ias directory and moved to the UIFbackup/console directory. The connector files need to be copied back to their original location.
2. Copy the console specific directories, the top level directory and the connector specific directories to the ias/APPS/console for the following folders:
   - constants
   - exceptions
   - ftp
   - gui
   - model
3. All the other directories, with all their included files, and the LocalDebug.class should be copied back.

Post-installation Issues

The following issues are described:

- Activating the Samples through NT/2000 or Solaris
- Copying Files in Webless Installation

Activating the Samples through NT/2000 or Solaris

After installing the iAS Enterprise Connector for CICS including the mainframe components, you can verify that your installation succeeded and run the CICS samples provided.

For more information on how to activate and run CICS samples, see Chapter 4 of the *iAS Enterprise Connector for CICS Developer's Guide.*

Copying Files in Webless Installation

In a webless installation, the web server and iPlanet Application Server reside on separate machine. In this configuration mode, the iAS Enterprise Connector for CICS installation procedure will not automatically copy the index.html file pertaining to connector samples to web server document root directory.

To run the samples shipped with the iAS Enterprise Connector for CICS, an index.html file must be copied manually from the iPlanet Application Server instance to web server instance.

To Copy Files to Run the Program Samples

1. Create the `cicsSamples` directory under the web server document root directory.

   For example, `<iWS installdir>/docs/cicsSamples`.

2. Copy the `index.html` file from `<iASrootdir>/ias/APPS/adapters/htmldocs/cics/cicsSamples` directory to `<iWS installdir>/docs/cicsSamples` directory on web server machine.

Chapter 3

Installing the Mainframe Component

After you install the iPlanet Application Server Enterprise Connector for CICS you need to install the mainframe component.

This chapter describes the following topics:

- Overview
- Transferring Libraries to the Mainframe
- Configuring CICS on the Mainframe
- Installing and Configuring the Relay
- Installing the PhoneBook Sample Application

Overview

You can connect the CICS Connector to the IBM mainframe in either of these ways:

- via TCP/IP: either using IBM or CA (TCPaccess)

  The TCP/IP must be one of the versions currently supported by either IBM or CA.

NOTE TCPaccess is the current name of Interlink.

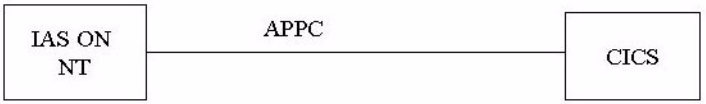
The iPlanet Application Server can be installed on Windows NT/2000 or Solaris and is connected via TCP/IP to a Relay component on OS/390 (MVS). The Relay communicates over the APPC with CICS. See Figure 3-1.

Figure 3-1 Installing the Connector via TCP/IP

- via Direct APPC/SNA

  The iPlanet Application Server can be installed on Windows NT/2000 and is connected via direct APPC connection to CICS on OS/390 (MVS). See Figure 3-2.

Figure 3-2 Installing the Connector via Direct APPC Using SNA



When you complete the installation, either via TCP/IP or via direct APPC using SNA installation, the PhoneBook sample application is available for end-to-end installation verification of the *iPlanet Application Server Enterprise Connector for CICS*. See "Installing the PhoneBook Sample Application".

Transferring Libraries to the Mainframe

The CICS components are supplied as downloaded PC files on the iPlanet Application Server Enterprise Connector for CICS installation media. You must upload the required Partitioned Data Sets (PDS) to MVS as part of the mainframe installation using either FTP or a 3270 emulator file-transfer utility. If TCP/IP is not installed, follow the instructions in "Uploading MVS Library Files Using 3270 File Transfer".

Uploading MVS Library Files via TCP/IP

If you have TCP/IP configured on your mainframe you can use the installation program *mfinst* that is described in the following sections.

Make sure that the FTP on the mainframe allocate data sets structured as follows:
`lrecl=80 blksize=3120 recfm=fb.`

If the allocated data sets are not structured as described above allocate four (or five) data sets on MVS with `1recl=80, blksize=3120, recfm=fb.`

For example:

- `IAS.CICS.LOAD.XMIT`
- `IAS.CICS.INSTALL.XMIT`
- `IAS.RELAY.INSTALL.XMIT`
- `IAS.RELAY.LOAD.XMIT`
- `IAS.CICS.DBRM.XMIT (Only needed for running the TELCO sample described in Appendix A, “Operating the TELCO Sample”)`

To Upload the Library Files on Windows NT/2000

- Run the `mfinst.bat` installation utility from the following directory:

  `\iPlanet\iAS6\APPS\adapters\cics\backend`

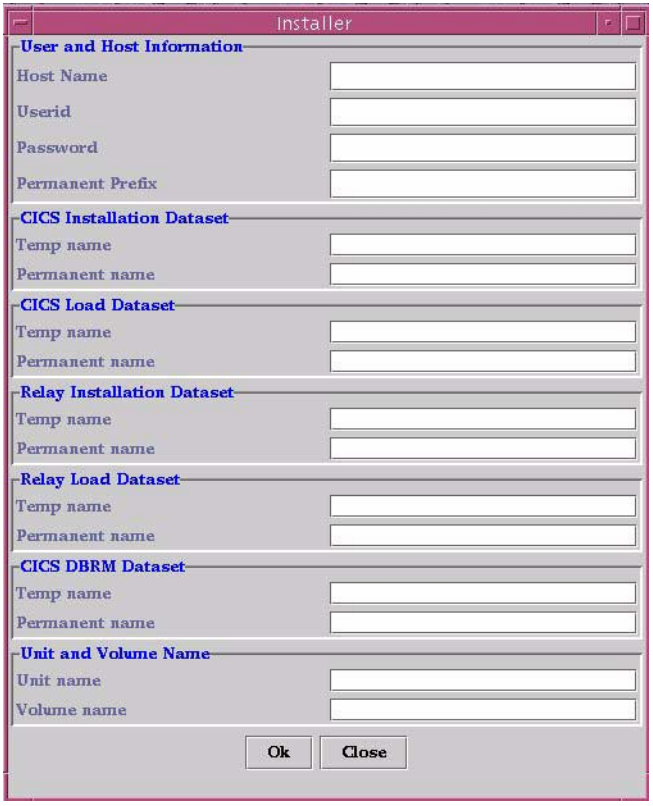
To Upload the Library Files on Solaris

- Run the file `mfinst.sh` installation utility from the following directory:

  `<instdir>/iAS6/ias/APPS/adapters/cics/backend`

  The Installer Dialog Box appears. See Figure 3-3.

Figure 3-3 Installer Dialog Box



The Installer Dialog Box contains information about the following fields:

- User and Host Information, see Table 3-1
- CICS Installation Dataset, see Table 3-2
- CICS Load Dataset, see Table 3-3
- Relay Installation Dataset, see Table 3-4
- Relay Load Dataset, see Table 3-5
- CICS DBRM, see Table 3-6
- Unit and Volume Name, see Table 3-7

User and Host Information

Table 3-1 lists names and IDs used to connect to the CICS system.

Table 3-1 User and Host Information

Variable	Definition
Host Name	Host name of the mainframe.
Userid	The administrator's ID on the mainframe.
Password	Password of administrator's ID
Permanent Prefix	The dataset prefix for the installation libraries. This prefix affects the permanent names of the CICS Installation Dataset, the CICS Load Dataset, the Relay Installation Datataset, the Relay Load Dataset, and the CICS DBRM dataset.

NOTE All fields except Unit and Volume Name will be automatically filled in after you enter the Permanent Prefix and press Return.

CICS Installation Dataset

The CICS Installation Dataset contains jobs for the CICS installation. Table 3-2 lists variables and definitions for the CICS Installation Dataset.

Table 3-2 CICS Installation Dataset

Variable	Definition
Temp Name	The temporary name assigned to the installation dataset during upload.
Permanent name	The permanent name assigned to the installation dataset. The prefix of this name will be the Permanent prefix defined in the User and Host information.

CICS Load Dataset

The CICS Load Dataset contains modules for the CICS connector. Table 3-3 lists variables and definitions for the CICS Load Dataset parameters.

Table 3-3 CICS Load Dataset

Variable	Definition
Temp Name	The temporary name assigned to the CICS Load dataset during upload.
Permanent name	The permanent name assigned to the CICS Load dataset.

Relay Installation Dataset

The Relay Installation Dataset contains jobs for the Relay installation. Table 3-4 lists and defines parameters for the Relay Installation Dataset.

Table 3-4 Relay Installation Dataset

Variable	Definition
Temp Name	The temporary name assigned to the Relay Installation dataset during upload.
Permanent name	The permanent name assigned to the Relay Installation dataset.

Relay Load Dataset

The Relay Load Dataset contains modules for the component. Table 3-5 lists and defines parameters for the Relay Load Dataset.

Table 3-5 Relay Load Dataset

Variable	Definition
Temp Name	The temporary name assigned to the Relay Load dataset during upload
Permanent name	The permanent name assigned to the Relay Load dataset.

CICS DBRM

The CICS DBRM is the Data Base Request Module library. Table 3-6 lists and defines parameters for the CICS DBRM..

Table 3-6 CICS DBRM

Variable	Definition
Temp Name	The temporary name assigned to the CICS DBRM dataset during upload.
Permanent name	The permanent name assigned to the CICS DBRM dataset.

Unit and Volume Name

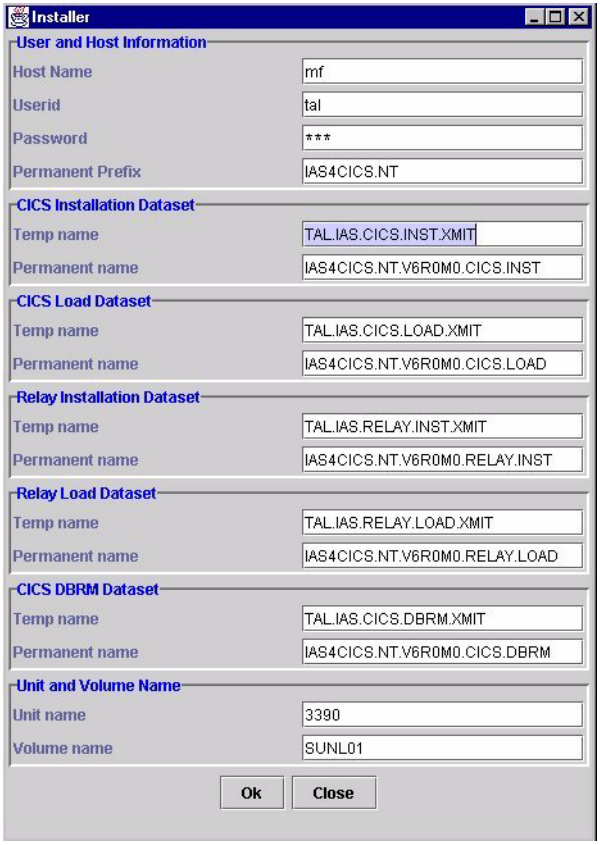
The Unit and Volume Name lists the assigned names for the Unit and Volume used.

Table 3-7 Unit and Volume Name

Variable	Definition
Unit name	The unit of the volume; i.e. 3390.
Volume name	The disk where the libraries are installed.

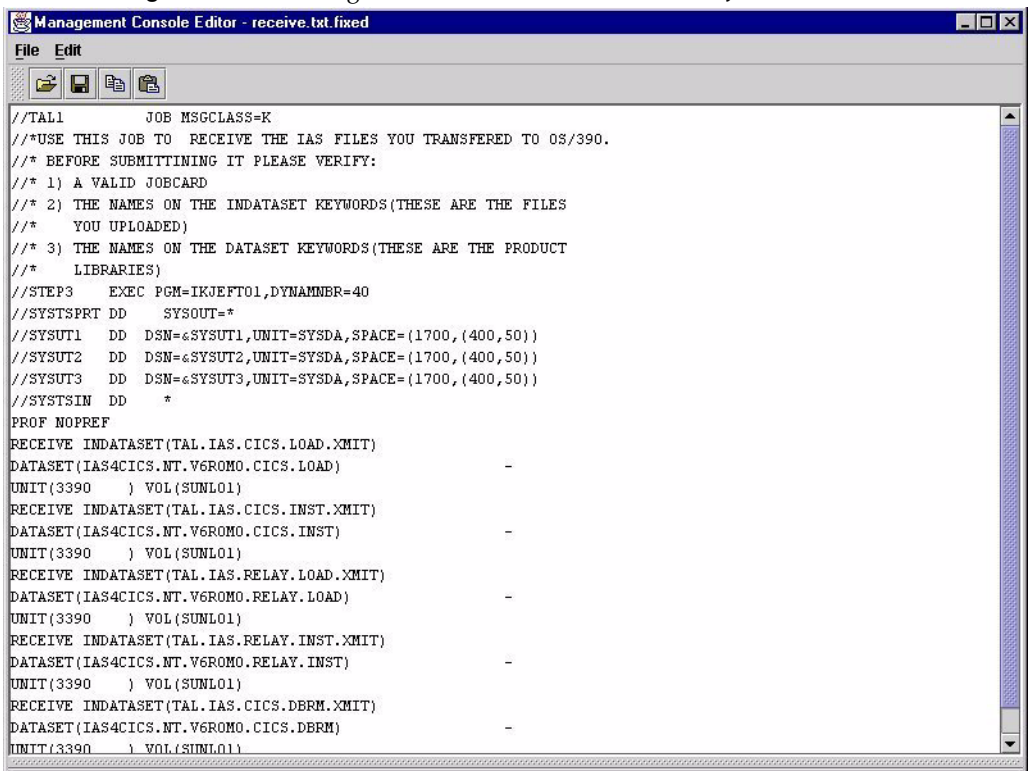
Figure 3-4 shows an example of a completed installer dialog box.

Figure 3-4 Completed Installer Dialog Box.



3. Press OK to continue. Figure 3-5 displays the generated JCL file.

Figure 3-5 Management Console Editor - Generated JCL



The JCL editor modifies the JCL to fit your mainframe environment according to the instructions as viewed in the screen capture above. All of the JCL file can be changed, except for the job name. To apply the changes, choose File and then Save before submitting the job.

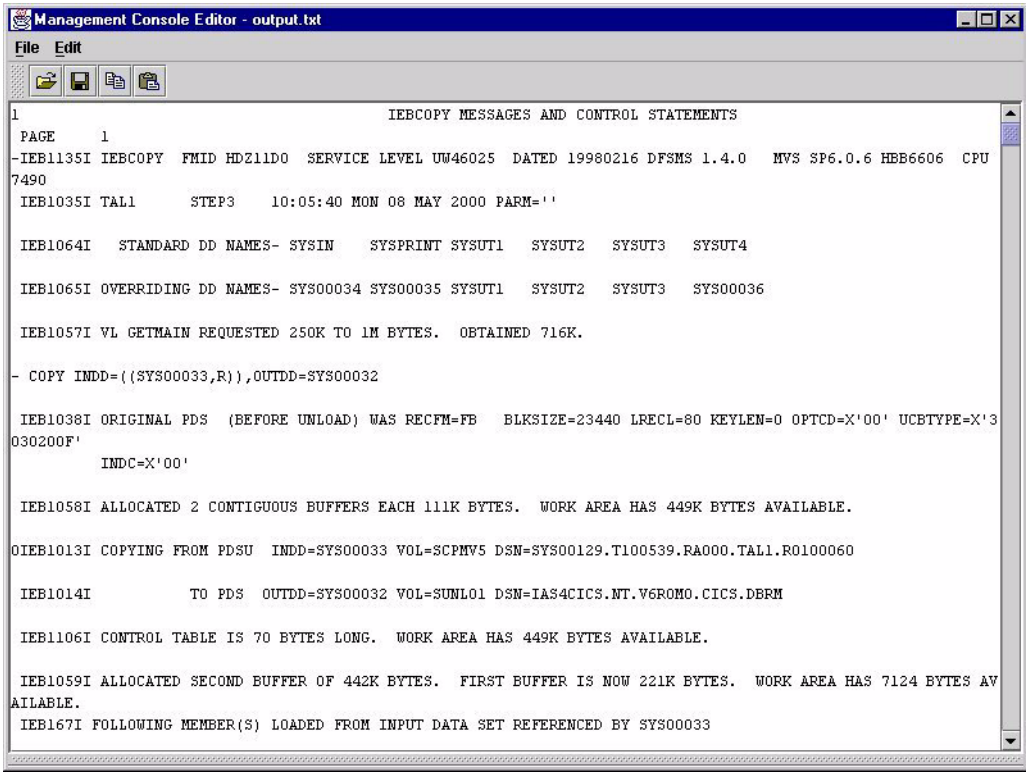
To Submit JCL for Execution

1. Select File>Exit from the menu bar.

   The utility will begin to FTP the files to the mainframe, and submit the JCL file to extract the files into permanent libraries.

   Once a job (JCL) finishes, outputs from the job are displayed as shown in Figure 3-6.

Figure 3-6 Management Console Editor: Outputs from job (JCL)



Review the job output.

2. Select File>Exit to close the window.
3. Click close to close the installation dialog box.
4. Go to Configuring CICS on the Mainframe.

Uploading MVS Library Files Using 3270 File Transfer

The following procedure applies to users that use a 3270 emulation software and its file transfer capability. This procedure details the steps for transfering MVS library files.

To Transfer Libraries to MVS

Step 1 - Allocate four data sets on MVS with `lrecl=80, blksize=3120, recfm=fb.`

For example:

- `IAS.CICS.LOAD.XMIT`
- `IAS.CICS.INSTALL.XMIT`
- `IAS.RELAY.INSTALL.XMIT (TCP/IP only)`
- `IAS.RELAY.LOAD.XMIT (TCP/IP only)`

Step 2 - Use the 3270 emulation program to transfer the following files to the respective MVS data sets. Use the MVS/TSO binary transfer type and fixed-file option.

- `LOAD.XMT`
- `INSTALL.XMT`
- `RELINST.XMT (TCP/IP only)`
- `RELLOAD.XMT (TCP/IP only)`

These files are found in

`<instdir>\IAS\APPS\adapters\cics\backend`

on Windows NT/2000.

Step 3 - Issue the following TSO command:

```
RECEIVE INDATASET(IAS.CICS.LOAD.XMIT)
```

Step 4 - At the prompt, enter the following:

```
DATASET(IAS4CICS.V6R5M0.CICS.LOAD)
```

where `IAS4CICS.V6R5M0.CICS.LOAD` is the name of the product load library.

Step 5 - Issue the following TSO command:

```
RECEIVE INDATASET(IAS.CICS.INSTALL.XMIT)
```

Step 6 - At the prompt, enter the following:

```
DATASET(IAS4CICS.V6R5M0.CICS.INST)
```

where `IAS4CICS.V6R5M0.CICS.INST` is the name of the installation library.

Step 7 - Issue the following TSO command: (TCP/IP only)

```
RECEIVE INDATASET(IAS.RELAY.INSTALL.XMIT)
```

Step 8 - At the prompt, enter the following: (TCP/IP only)

```
DATASET(IAS4CICS.V6R5M0.RELAY.INST)
```

where `IAS4CICS.V6R5M0.RELAY.INST` is the name of the installation library for the RELAY component.

Step 9 - Issue the following TSO command: (TCP/IP only)

```
RECEIVE INDATASET(IAS.RELAY.LOAD.XMIT)
```

Step 10 - At the prompt, enter the following: (TCP/IP only)

```
DATASET(IAS4CICS.V6R5M0.RELAY.LOAD)
```

where `IAS4CICS.V6R5M0.RELAY.LOAD` is the name of the load library for the `Relay` component.

Configuring CICS on the Mainframe

You need to configure the components to install the connection to the mainframe computer. This section contains a sumamry of the configuration steps followed by a section with the detailed procedures.

Summary of Procedure To Configure the CICS Components

Step 1- Add the Product Load Library to CICS DFHRPL List.

Step 2- Ensure APPC Logmode Entry is Present.

Step 3 - Verify CICS Applid Definitions

NOTE Direct APPC only.

Step 4 - Define a Remote PU for each NT/2000 Server Platform Connected Directly to CICS via APPC

Step 5 - Changing Installation Parameters' Default Names (Optional).

Step 6 - Update the CICS Resource Definitions by Running the Supplied CICSDEF.

Step 7 - Install a Resource Group for the APPC Connection.

Details of Procedure To Configure the CICS Components

Step 1- Add the Product Load Library to CICS DFHRPL List

Add the product load library to the CICS DFHRPL list in the CICS startup JCL.

Code Example 3-1 Sample VTAM Major Node for CICS

```
Startup/JCL//DFHRPL DDDSN=
          DD  DSN='
.......
 .......
 DD DSN='IAS4CICS.V6R5M0.CICS.LOAD'.DISP=SHR*
```

Step 2- Ensure APPC Logmode Entry is Present

NOTE If you do not have special logmode needs you can use the IBM supplied logmode APPCSNA and go directly to "Step 3 - Verify CICS Applid Definitions".

Code Example 3-2 displays a MODE entry in the MODE TABLE. The same Logmode must be associated with the CICS VTAM definition.

If you perform the CICS command: `cemt I mode`(appcmode) you should get a positive result. If `NOT FOUND` is displayed, see "Step 6 - Update the CICS Resource Definitions by Running the Supplied CICSDEF" for the logmode that must be added to the CICS.

Code Example 3-2 Creating the VTAM mode table MTAPPC

```
//MODETAB JOB
//* ANY SITE PROCEDURE FOR ASSEMBLY AND LINK CAN BE USED
//* THIS SAMPLE IS USING HIGH LEVEL ASSEMBLER
//* IN LATEST OS/390 VERSIONS, IEWL WILL INVOKE THE DFSMS BINDER
//ASM EXEC PGM=ASMA90,REGION=5048K,
// PARM='OBJECT,NODECK,LIST,XREF(SHORT)'
//SYSLIB DD DSN=SYS1.SISTMAC1,DISP=SHR
//SYSUT1 DD UNIT=SYSDA,SPACE=(CYL,(10,5))
```

Code Example 3-2 Creating the VTAM mode table MTAPPC

```
//SYSIN DD *
MTAPPC MODETAB
APPCMODE MODEENT LOGMODE=APPCMODE,                                    *
               FMPROF=X'13',                                          *
               TSPROF=X'07',                                          *
               PRIPROT=X'B0',                                         *
               SECPROT=X'B0',                                         *
               COMPROT=X'D0B1',                                       *
               RUSIZES=X'8585',                                       *
               PSERVIC=X'060200000000000000000300'
         MODEEND
       END
//SYSLIN DD DSN=&&LOADME,DISP=(,PASS),UNIT=SYSDA,
// SPACE=(TRK,(30,10),RLSE)
//SYSPRINT DD SYSOUT=*
//LKED EXEC PGM=IEWL,PARM='MAP,LET,LIST',REGION=912K,
// COND=(8,LT,ASM)
//SYSLIB DD DSN=SYS1.MACLIB,DISP=SHR
//SYSLMOD DD DISP=SHR,DSN=SYS1.VTAMLIB
//SYSUT1 DD UNIT=SYSDA,SPACE=(CYL,(3,2)),DSN=&SYSUT1
//SYSPRINT DD SYSOUT=*
//SYSLIN DD DSN=&&LOADME,DISP=(OLD,DELETE)
// DD *
 NAME MTAPPC(R)
```

Step 3 - Verify CICS Applid Definitions

Code Example 3-3 is a sample VTAM definition for CICS. In most cases, CICS is already defined and running.

Code Example 3-3 Sample VTAM major node for CICS

```
SUNCICS2 VBUILD TYPE=APPL                APPLICATION MAJOR NODE
A06CICS2 APPL  EAS=160,                ESTIMATED CONCURRENT SESSIONS   *
               ACBNAME=A06CICS2,       APPLID FOR ACB                  *
               MODETAB=MTAPPC,                                         *
               PARSESS=YES,                                            *
               SONSCIP=YES,                                            *
               AUTH=(ACQ,VPACE,PASS)   CICS CAN ACQUIRE & PASS TMLS
```

If you choose to use logmode APPCSNA, `MODETAB` is not required.

NOTE Do not add APPC=YES to the CICS VTAM definition.

Step 4 - Define a Remote PU for each NT/2000 Server Platform Connected Directly to CICS via APPC

If your application server is connected directly to CICS (using APPC connections), define the remote PU (the application server machine) to VTAM and CICS.

Refer to Code Example 3-4.

Code Example 3-4 Sample PU and LU6.2 Definition for Windows NT/2000

```
*/*
SWSHMUEL VBUILD TYPE=SWNET
PUSHMUEL   PU PUTYPE=2,ADDR=C1,MAXDATA=1456,                          *
              CPNAME=SHMUELNT
SHMUELNT     LU LOCADDR=0
```

Step 5 - Changing Installation Parameters' Default Names (Optional)

The default names of the installation parameter may be changed but it is not recommended.

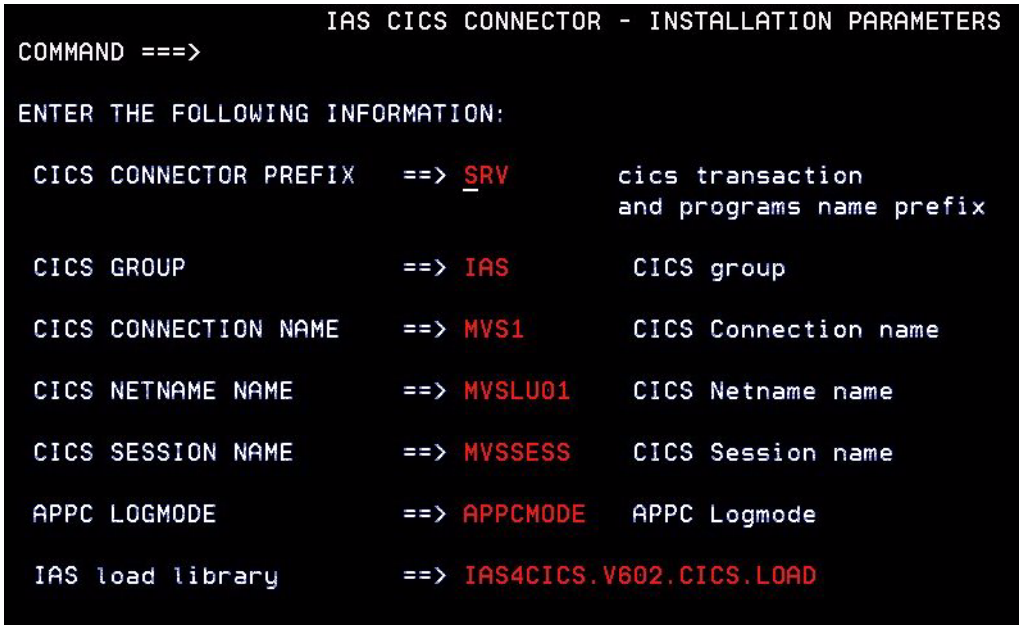
To Change the CICS Installation Parameters

1. Run the INST REXX located in the CICS installation library.

   The REXX prompts you for the library name.

2. Type in the library name, without quotes, and press enter. Figure 3-7 is displayed.

Figure 3-7 Installation Parameters



3. If you change "*CICS CONNECTOR PREFIX*" from SRV then you must also change "*IAS LOAD LIBRARY*" to your CICS load library. It must be the same as the "*CICS LOAD DATASET PERMANENT NAME*" shown in Figure 3-4.

   This is the library of the CICS modules that you want to change.

4. You will get the following job.

Code Example 3-5 ZAP job for Mainframe

```
//IASJOB JOB (PLEASE EDIT JOB CARD)
//ZAP1     EXEC PGM=AMASPZAP,REGION=4M
//SYSPRINT DD SYSOUT=*
//SYSLIB   DD DISP=SHR,DSN=IAS62.V6R2M0.CICS.LOAD
//SYSIN    DD *
NAME SRVXLINK SRVXLINK
VER   0700 E2D9E5
REP   0700 D2D2D2
//ZAP2     EXEC PGM=AMASPZAP,REGION=4M
//SYSPRINT DD SYSOUT=*
//SYSLIB   DD DISP=SHR,DSN=IAS62.V6R2M0.CICS.LOAD
//SYSIN    DD *
NAME SRVRSRPC SRVSEC
```

Code Example 3-5 ZAP job for Mainframe

```
VER   01A0 E2D9E5
REP   01A0 D2D2D2
//TSO  EXEC PGM=IKJEFT01,DYNAMNBR=30,REGION=4M,COND=(0,NE)
//SYSPRINT DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
 PROFILE NOPREFIX WTPMSG MSGID
RENAME IAS62.V6R2M0.CICS.LOAD(SRVXLINK) +
 IAS62.V6R2M0.CICS.LOAD(KKKXLINK)
RENAME IAS62.V6R2M0.CICS.LOAD(SRVRSRPC) +
 IAS62.V6R2M0.CICS.LOAD(KKKRSRPC)
RENAME IAS62.V6R2M0.CICS.LOAD(SRVSCURE) +
 IAS62.V6R2M0.CICS.LOAD(KKKSCURE)
```

5. Edit the job card.
6. Submit the job.
7. Verify that the job executed successfully.

Step 6 - Update the CICS Resource Definitions by Running the Supplied CICSDEF

Update the CICS resource definitions by running the supplied CICSDEF from the CICS installation library.

Refer to Code Example 3-6.

When using direct APPC connection from Windows NT/2000, refer to Code Example 3-7 as a definition for CICS connection and sessions.

Code Example 3-6 CICSDEF to Define CICS Components

```
//CICSDEF  JOB CLASS=A,MSGCLASS=K
//*
//* SAMPLE JOB TO DEFINE THE IAS FOR CICS  COMPONENTS TO CICS
//* CHECK THE DSN= PARAMETERS FOR THE STEPLIB AND DFHCSD  DD CARDS
//* CHANGE 'CICSLIST' TO THE CICS STARTUP LIST NAME
//* SEE NOTES FOR EACH SECTION BELOW (FILES, PROGRAMS, ETC.)
//*
//CSDUP   EXEC PGM=DFHCSDUP
//STEPLIB  DD DSN=CICS.CICS.SDFHAUTH,DISP=SHR
//DFHCSD   DD DSN=CICS.DFHCSD,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
```

Code Example 3-6 CICSDEF to Define CICS Components *(Continued)*

```
***
***
*** DEFINITIONS FOR TRANSACTIONS
***
  DEFINE TRANSACTION(SRVX) GROUP(IAS) PROGRAM(SRVXLINK)
         DESCRIPTION(IAS X-LINK UNIT)
***
*** DEFINITIONS FOR PROGRAMS
***
  DEFINE PROGRAM(SRVRQRPC) GROUP(IAS) LANGUAGE(ASSEMBLER)
  DEFINE PROGRAM(SRVSCURE) GROUP(IAS) LANGUAGE(ASSEMBLER)
  DEFINE PROGRAM(SRVXLINK) GROUP(IAS) LANGUAGE(ASSEMBLER)
         DESCRIPTION(IAS X-LINK UNIT)
***
*** DEFINITIONS FOR APPC CONNECTIONS AND SESSIONS
***
***   THIS IS A SAMPLE DEFINITION, DEFINING THE CONNECTION
***   AND SESSIONS TO APPC/MVS. IT ASSUMES THE SAME NAMES
***   AS IN THE SUPPLIED (ABOVE) SAMPLES. IF ANY WERE
***   MODIFIED, THESE MUST BE REFLECTED BELOW.
***
TCP/IP ONLY
***   THE CONNECTION AND SESSIONS FOR APPC/MVS ARE USED IN CASE IAS
***   IS CONNECTED OVER TCP/IP TO RELAY, AND RELAY USES APPC/MVS
***   TO CONNECT TO CICS.
***   IF YOU ARE CONNECTING IAS ON NT USING DIRECT APPC CONNECTION
***   TO CICS REFER TO SAMPLE MEMBER CICSDEF3 IN THIS LIBRARY.
***
  DEFINE CONNECTION(MVS1) GROUP(IAS)
         NETNAME(MVSLU01) ACCESSMETHOD(VTAM) PROTOCOL(APPC)
         SINGLESESS(NO) INSERVICE(YES) AUTOCONNECT(YES)
         DESCRIPTION(SAMPLE CONNECTION, TO APPC/MVS)
  DEFINE SESSION(MVS1SESS) GROUP(IAS)
         CONNECTION(MVS1) MODENAME(APPCMODE) PROTOCOL(APPC)
         MAXIMUM(10,5)  AUTOCONNECT(YES)
         DESCRIPTION(SAMPLE CONNECTION, TO APPC/MVS)
***
*** ADD THE IAS GROUP, WITH THE DEFINITIONS, TO THE CICS STARTUP LIST
*** OR INSTALL GROUP(IAS) TO PUT DEFINITIONS IN EFFECT
***
  ADD   GROUP(IAS) LIST(CICSLIST)
/*
//
```

Step 7 - Install a Resource Group for the APPC Connection

NOTE The session definition contains the MODENAME which is also present in mode tables associated with APPC/MVS and the CICS definitions in VTAM. The same mode name should be used in all three instances.

Code Example 3-7 can only be used for a direct APPC connection from Windows NT/2000, and demonstrates definitions for CONNECTION and SESSIONS parameters in CICS.

Code Example 3-7 CICSDEF3

```
//CICSDEF3 JOB
//*
//CSDUP   EXEC PGM=DFHCSDUP
//STEPLIB  DD DSN=CICS.CICS.SDFHAUTH,DISP=SHR
//DFHCSD   DD DSN=CICS.DFHCSD,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
***
*** DEFINITIONS FOR IAS/NT USING DIRECT APPC.
*** THESE DEFINITIONS MUST BE INSTALLED IN CICS RESOURCE DEFINITION
*** IN ADDITION TO, OR INSTEAD OF, CONNECTION AND SESSIONS FOR APPC/MVS
*** ENLISTED IN CICSDEF SAMPLE.
***
  DEFINE CONNECTION(SKNT) GROUP(IASDAPPC)
         NETNAME(SHMUELNT) ACCESSMETHOD(VTAM) PROTOCOL(APPC)
         SINGLESESS(NO) INSERVICE(YES) AUTOCONNECT(YES)
         DESCRIPTION(SAMPLE DIRECT/APPC CONNECTION TO IAS/NT)
  DEFINE SESSIONS(SKNTAPPC) GROUP(IASDAPPC)
         CONNECTION(SKNT) MODENAME(APPCMODE) PROTOCOL(APPC)
         MAXIMUM(10,5)  AUTOCONNECT(YES)
         DESCRIPTION(SAMPLE DIRECT CONNECTION TO IAS/NT)
***
*** DEFINITIONS ARE PUT IN EFFECT BY ADDING TO CICS STARTUP LIST AS
*** DEMONSTRATED HERE, OR BY INSTALLING GROUP CONTAINING THESE DEFS
***
  ADD GROUP(IASDAPPC) LIST(CICSLIST)
/*
//
```

Installing and Configuring the Relay

Installing and configuring the Relay is only necessary when you use TCP/IP to connect to the mainframe.

When using the Relay, configure the environment for the following:

- VTAM
- TCP/IP
- APPC/MVS
- Security

Proceed in the following order:

Step 1 - Create an Application Table Dataset for the Relay Component

Step 2 - Configure APPC/MVS

Step 3 - Create a TCP/IP Listener Started Task

Step 4 - Verify the Security Server Definition

Step 6- Operate the Listener

Step 1 - Create an Application Table Dataset for the Relay Component

You must create an application dataset for the Relay component.

To Create an Application Table Dataset for the Relay Component

1. Run the DEFFILES REXX located in the Relay installation library.

   The REXX prompts you for the entries. See Step 2 through Step 5, below.
2. Enter the application table name without quotes.

   Specify a name for the relay's application table. For example:

   IAS.APPLTAB.RELAY
3. Enter the VOLSER (Volume Serial Name).

   Specify the name of the DASD on which this file should be created.
4. Enter unit type, or press Enter for 3390.

   Specify the unit type.

5. Enter MVS/APPC logmode.

   Enter the logmode that should be used to access CICS. The supplied name is APPCMODE.

   The message *"Application Table Created Successfully"* appears when the table is created.

Step 2 - Configure APPC/MVS

This procedure is required only if APPC/MVS is not already configured on your system. If it is configured on the system, it is recommended that you read through these instructions and compare them to the existing configuration on your system. You can use either one of the samples provide with the iAS Enterprise Connector for CICS to configure APPC/MVS.

To Configure APPC/MVS

1. If APPC/MVS is not installed on your system, you can adapt and use the sample code given in Code Example 3-8 or the sample definition for APPC/MVS base LU given in Code Example 3-9.

   The member APPC/MVS in the Relay Installation library contains VTAM definitions for the base APPC/MVS LU.

Code Example 3-8 Library Member APPCMVS

```
*
* SAMPLE MAJOR NODE FOR APPC/MVS
*
APPCMVS  VBUILD TYPE=APPL
MVSLU01  APPL  ACBNAME=MVSLU01,                                        *
               APPC=YES,                                               *
               AUTOSES=10,                                             *
               DDRAINL=NALLOW,                                         *
               DMINWNL=20,                                             *
               DMINWNR=20,                                             *
               DRESPL=NALLOW,                                          *
               DSESLIM=40,                                             *
               EAS=509,                                                *
               MODETAB=MTAPPC,                                         *
               SECACPT=CONV,                                           *
               SRBEXIT=YES,                                            *
               VERIFY=NONE,                                            *
               VPACING=2
```

NOTE The iPlanet Application Server Enterprise Connector for CICS does not require ASCH.

Code Example 3-9 APPCPM00 Member

```
/* LIB: SYS1.PARMLIB(APPCPM00)                                    */
LUADD ACBNAME(MVSLU01) BASE NOSCHED
```

2. Activate the APPC/MVS major node by using the following command:

   ```
   V NET,ID=APPCMVS,ACT
   ```

3. Using the MVS commands, start APPC/MVS by using the following command:

   ```
   S APPC,SUB=MSTR,APPC=00
   ```

Step 3 - Create a TCP/IP Listener Started Task

The Listener listens to a TCP/IP port. The following sections describe both the prerequisites and the procedure for configuring and operating the Listener.

NOTE Because the Listener and the Relay reside in the same address space, when referring to the Listener and Relay, the reference is to the same component.

The TCP/IP Listener can be executed as an MVS Started Task, or as a batch job. It is recommended to run it as an MVS Started Task.

To Create and Configure the Listener

1. Copy the member LISTEN, (the sample procedure in Relay installation library), into a system procedure library, for example,

   ```
   SYS1.PROCLIB.
   ```

2. Update the following PROC statement parameters:

Table 3-8 PROC Statement Parameters

TCPIP	Name of the TCP/IP address space. TCPIP is the default.
PORT	The same (Relay) port specified in Table 2-1.
MAXTASK	Maximum number of concurrent tasks. 20 tasks is the default.
PROGRAM	Name of the program to activate for incoming requests must be 'SRVRLNS'.

3. Ensure that the STEPLIB points to the load library.

   The load library *must* be APF authorized.

   Place an entry for the load library in your paramfile library member IEAAPFxx or PROGxx, depending on your site's standards.

4. Ensure that the SRVAPLF DD card points to the application table.

5. Verify the SYSOUT class on the CPEOUT, CPEOUTRS, CPEOUTRO DD cards.

   Refer to Code Example 3-10.

Code Example 3-10 Listen PROC

```
//LISTEN PROC TCPIP=TCPIP,PORT=47000,MAXTASK=20,PROGRAM=SRVRLNS

//*********************************************************************
//* THIS PROCEDURE CAN BE USED FOR A TCP/IP LISTNER.
//* MOST COMMONLY, IT IS USED AS A RELAY BETWEEN TCP/IP AND APPC.
//* IT CAN BE RUN AS A STARTED TASK OR A JOB.
//*
//* BEFORE USING IT PLEASE PERFORM THE FOLLOWING CHANGES:
//* 1. CHECK THE NAME OF THE STEPLIB. NOTE THAT THIS LIBRARY HAS TO BE
//*    AUTHORIZED.
//* 2. CHECK THE NAME OF THE APPLICATION TABLE ON SRVAPLF
//* 3. CHECK THE SYSOUT CLASS ON CPEOUT AND SYSUDUMP.
//* 4. UPDATE THE PARAMETERS ON THE PROC STATEMENT :
//*  TCPIP =  THE NAME OF THE TCP/IP ADDRESS SPACE (DEFAULTS TO TCPIP)
//*  PORT  =  AN AVAILABLE PORT NUMBER (DEFAULS TO 47000)

//*  MAXTASK = THE  MAXIMUM NUMBER OF CONCURRENT TASKS (DEFAULTS TO 20)
//*  PROGRAM = THE  NAME OF THE PROGRAM TO ACTIVATE ON INCOMING REQUEST
//*       - LEAVE BLANK TO PICK UP THE NAME IN THE APPLICATION TABLE
//*       - CODE 'SRVRLTA' FOR A RELAY
//LISTEN EXEC PGM=SRVLISPR,PARM='&TCPIP,&PORT,&MAXTASK,&PROGRAM',
//        REGION=0M
//STEPLIB   DD DSN=IAS4CICS.V6R5M0.RELAY.LOAD,DISP=SHR
//CPEOUT    DD SYSOUT=X,HOLD=YES
```

Code Example 3-10 Listen PROC *(Continued)*

```
//CPEOUTRQ  DD SYSOUT=X,HOLD=YES
//CPEOUTRS  DD SYSOUT=X,HOLD=YES
//CPESNAP   DD SYSOUT=X,HOLD=YES
//SYSUDUMP  DD SYSOUT=X,HOLD=YES
//SRVAPLF   DD DSN=IAS.APPLTAB.RELAY,DISP=SHR
//SRVTCPI   DD DUMMY
// PEND
```

Step 4 - Verify the Security Server Definition

Verify that the user ID assigned to the Listener has an OMVS segment.

Example: To add the listener, that is assigned to user-ID LISTEN, the OMVS segment should be added by issuing the following RACF command:

```
ADDUSER LISTEN OMVS(UID(1234...))
```

Step 5 (Optional) - Security Exit for the Listener

This exit can be used to perform any kind of security check. You can for example call RACROUT and verify that the user has access to a particular resource or check the IP address of the incoming request against a list.

For details see

Step 6- Operate the Listener

The following description of how to start the Listener assumes that `LISTEN` is the name of the started task.

Use the following procedures to start, stop, and restart the Listener, and to determine and free the connection port.

To Start the Listener

- Issue the MVS command: `S LISTEN`

NOTE LISTEN can be included in the TCP/IP automatic startup list (AUTOLOG).

To Stop the Listener

- Issue the MVS command: `P LISTEN`

To Stop the Listener After a Fixed Amount of Time

- Issue the MVS command: `F LISTEN,SHUTDOWN` *nnn*

  where *nnn* is the number of seconds (default is 10 seconds) and allows open conversations to end.

In early releases of TCP/IP, even though the Listener was stopped, its port may not be freed. In such cases, determine the connection identification and free the port.

To Determine the Connection Identification

- Issue the following command:

  ```
  NETSTAT ALLCONN (PORT xxxxx)
  ```

  where xxxxx is the port assigned to the Listener (Relay). Note the connection identification.

To Free the Port

- Issue the following command which requires the issuing user to have the proper security authorization:

  ```
  NETSTAT DROP yyyyy
  ```

  where yyyyy is the connection identification obtained in the preceding NETSTAT command.

Installing the PhoneBook Sample Application

The PhoneBook Sample installation is intended to verify that the iPlanet Application Server Enterprise Connector for CICS has been properly installed.

NOTE The TELCO sample application including its installation is described in the appendix. The TELCO sample requires DB2 for OS/390 and is not part of installation verification.

To Install the Sample PhoneBook Application on CICS

1. Edit the PBINIT member located in the CICS installation library according to the instructions in the member.
2. Submit the PBINIT member.

   Refer to Code Example 3-11.

Code Example 3-11 PhoneBook Sample in the CICS Environment

```
//PBINIT JOB MSGCLASS=K
//*===============================================================*
//*
//* THIS JOB CREATES THE VSAM FILE FOR THE IAS PHONEBOOK
//* DEMO. IT CREATES, INITIALIZES THE VSAM PHONEBOOK DATASET, AND
//* DEFINES IT TO CICS USING THE BATCH CSD UPDATE UTILITY.
//*
//*===============================================================*
//*   1.   DEFINE VSAM FILE
//*
//*  CHECK THE NAME OF THE LIBRARY IN THE SYSEXEC CARD, IT SHOULD BE
//*        THIS LIBRARY
//*  CHECK THE VSAM DATASET NAME(S) IN THE SYSTSIN LINE 1 AND 3
//*  CHANGE THE VOLSER (VVVVVV) IN SYSTSIN LINE 4
//*===============================================================*
//STEP3    EXEC PGM=IKJEFT01,DYNAMNBR=40
//SYSTSPRT DD    SYSOUT=*
//SYSUT1   DD  DSN=&SYSUT1,UNIT=SYSDA,SPACE=(1700,(400,50))
//SYSUT2   DD  DSN=&SYSUT2,UNIT=SYSDA,SPACE=(1700,(400,50))
//SYSUT3   DD  DSN=&SYSUT3,UNIT=SYSDA,SPACE=(1700,(400,50))
//SYSEXEC  DD  DSN=IAS4CICS.V6R5M0.CICS.INST,DISP=SHR
//INPUT DD *
ARIE      NOA       262       12324
FLINT     ORIT      260       4377
GALIL     MOTI      1234      12345
GALMIDI   MOTI      123       1234
GOLD      KEREN     258       52960
GRYNBAUM  TAL       251       52960
MICHEAL   SHATZ     234       43456
REICH     SHANY     123       1234
SHANI     RAKEFET   252       1234
SHIMON    RICKY     259       43788
SHORKEND  MIKE      123       1234
/*
//SYSTSIN  DD    *
DEL 'IAS.PHONBOOK.CICS'
PBLOAD
IAS.PHONBOOK.CICS
VVVVVV
```

Code Example 3-11 PhoneBook Sample in the CICS Environment *(Continued)*

```
3390
/*
//*==================================================================*
//*  2.   UPDATE CICS DEFINITIONS WITH PHONEBOOK APPLICATION DATA
//*
//*  CHECK THE DSN= PARAMETERS FOR THE STEPLIB AND DFHCSD DD CARDS
//*  CHANGE 'CICSLIST' TO THE CICS STARTUP LIST NAME
//*  SEE NOTES FOR EACH SECTION BELOW (FILES, PROGRAMS, ETC.)
//*
//*==================================================================*
//CSDUP   EXEC PGM=DFHCSDUP
//STEPLIB  DD DSN=CICS.CICS.SDFHLOAD,DISP=SHR
//DFHCSD   DD DSN=CICS.DFHCSD,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
***
*** DEFINITIONS FOR PHONEBOOK FILES
***
  DEFINE FILE(PBVSAM) GROUP(IASPHB)
         DSNAME(IAS.PHONBOOK.CICS) DISPOSITION(SHARE)
         ADD(YES) BROWSE(YES) DELETE(YES) READ(YES) UPDATE(YES)
         RECORDFORMAT(F) STATUS(ENABLED) OPENTIME(FIRSTREF)
         JOURNAL(01) STRINGS(10) DATABUFFERS(11)
         DESCRIPTION(IAS PHONEBOOK APPLICATION TABLE)
***
*** DEFINITIONS FOR TRANSACTIONS
***
***   THE TRANSID VALUES MAY BE CHANGED.
***
  DEFINE TRANSACTION(PHBK) GROUP(IASPHB) PROGRAM(PHONBOOJ)
         DESCRIPTION(IAS PHONEBOOK APPLICATION)
***
*** DEFINITIONS FOR PROGRAMS
***
  DEFINE PROGRAM(PHONBOOJ) GROUP(IASPHB) LANGUAGE(ASSEMBLER)
  DEFINE PROGRAM(PHONBOOX) GROUP(IASPHB) LANGUAGE(ASSEMBLER)
***
*** ADD THE IAS GROUP, WITH THE DEFINITIONS, TO THE STARTUP LIST
*** IF NEEDED (NOT DONE BEFORE
***
  ADD   GROUP(IASPHB) LIST(CICSLIST)
/*
```

To Operate the Sample PhoneBook Application on NT/2000

1. Select Start>Programs>iPlanet Application Server 6.5.
2. Select CICS Connector 6.5 - Sample Applications.

To Operate the PhoneBook Sample Application on Solaris

- Run Netscape or Microsoft browser with the following URL:

  http://<webserver instance name>/cicsSamples

Chapter 4

Managing Data

This chapter describes how to use the Management Console and the Repository Browser to manage and view your data.

This chapter contains information about the following topics:

- The Management Console
- The Repository Browser

The Management Console

Each connector has an interactive tool, the Management Console. The Management Console is typically used by the data source administrator and consists of two groups of functions:

- Tasks that are used to process and browse data and functions from Enterprise System metadata:
  - data mining
  - creating new data sources
  - editing data sources
- Mapping User Privileges that establishes relationships between the web and back-end users.
- The Listener Management enables the administrator to view and edit listener parameters, create new listeners and view listener statistics.

To Enter the CICS Management Console from Windows NT/2000

- Select Start>Programs>iPlanet Application Server 6.5 >CICS Connector 6.5-Management Console.

To Enter the CICS Management Console from Solaris

1. Enter the following directory:

   ```
   cd <iPlanet>/ias/APPS/bin
   ```

2. Enter the following command:

   ```
   cicsconsole.sh
   ```

   The CICS Management Console window is displayed. See Figure 4-1.

Figure 4-1 CICS Management Console

Management Console Functions

The Tasks tab is used for:

- Creating a New Data Source
- Editing a Data Source

The CICS User Management tab is used for:

- Mapping User Privileges

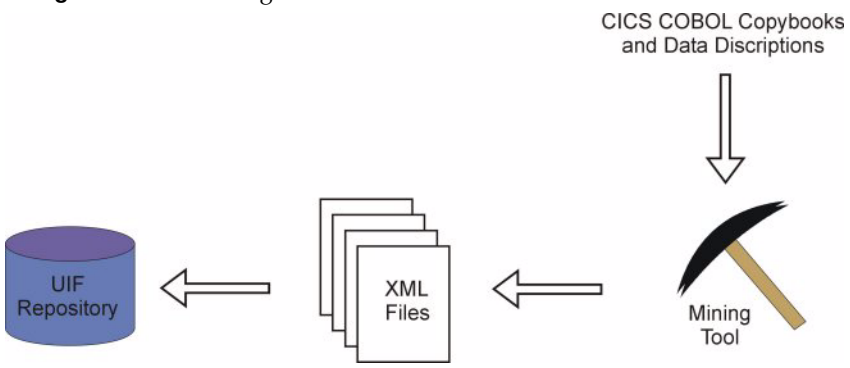
The Listener Management tab is used for:

- Listener Statistics
- Listener Management

The Data Mining Tool

The Data Mining Tool uses a common cross-platform format, eXtensible Markup Language (XML), to populate the repository by taking back-end information and converting it to XML. The XML is then stored in the UIF Repository in a specific structure that can be browsed when necessary.

Figure 4-2 Mining Tool



The Data Mining Tool allows you to extract functions and parameters from CICS. The iPlanet Application Server user can then utilize (call) these functions.

Accessing the Data Mining Tool

The Data Mining Tool in the *iPlanet Application Server Enterprise Connector for CICS* is accessed through the Management Console.

To Enter the Data Mining Tool

1. Enter the CICS Management Console as described in The Management Console.
2. In the Tasks tab, select the Data Mining icon.

   The Option Chooser window is displayed see Figure 4-3.

Figure 4-3 Option Chooser Window

You can select a local or remote Cobol file.

NOTE To convert a local Cobol File, continue to To Choose a Cobol File (local).

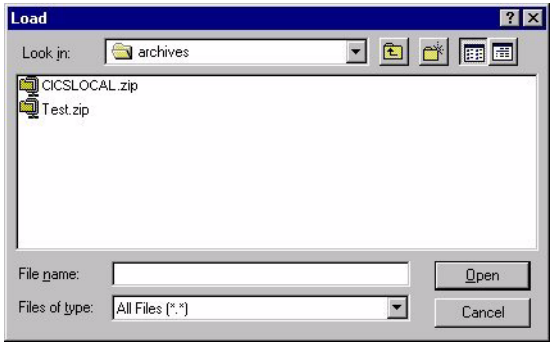
To convert a remote Cobol File, go to To Choose a Cobol File (Remote).

To Choose a Cobol File (local)

1. In the Option Chooser window, select the option Convert local Cobol File, and click OK.

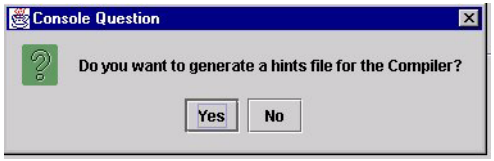
   A window is displayed enabling you to select and load a local file. See Figure 4-4.

Figure 4-4 Load Window



2. Select and load a file and click Open. A Console Question window is displayed. See Figure 4-5.

Figure 4-5 Console Question



3. Click Yes to generate a hints file for the Compiler, or No.

   A Hints file informs the Cobol data mining tool how to treat redefined variables. For more information, consult a Cobol source book.

   If Yes, an editor screen opens with the hints file displayed.

   The Cobol file is converted to XML and loaded into the repository.

To Choose a Cobol File (Remote)

1. In the Option Chooser window, (Figure 4-3), select the option Convert remote Cobol File, and click OK.

   The Get Remote File window is displayed. See Figure 4-6.

Figure 4-6 Get Remote File Window

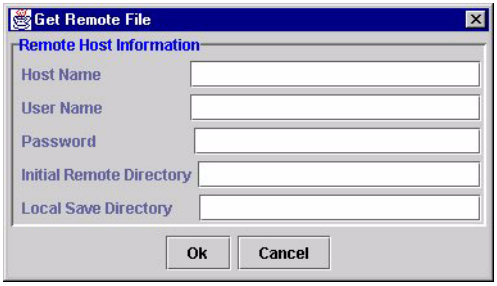


Table 4-1 describes the fields for the Remote Host Information where the copybook file is downloaded from using FTP.

2. Enter in the information required.

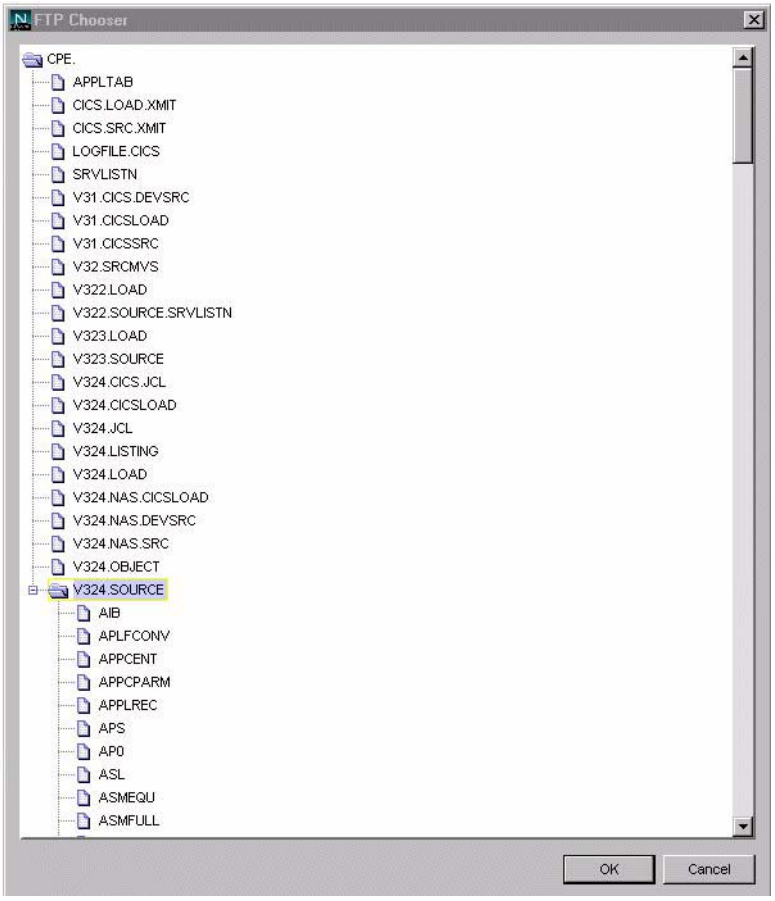
Table 4-1 Remote Host Information

Variable	Definition
Host Name	Name of the host computer where the remote file is located.
User Name	Name of the host user ID.
Password	Password of the host user ID.
Initial Remote Directory	Name of the directory where you start browsing the mainframe files.
Local Save Directory	Name of the local directory where the file is saved.

3. Click OK.

   The FTP Chooser window is displayed, see Figure 4-7. The file is converted to XML format, and loaded into the Repository.

Figure 4-7 FTP Chooser



4. Select a file (or files) from the tree and click OK. The file(s) will be downloaded, data-mined and loaded into the repository.

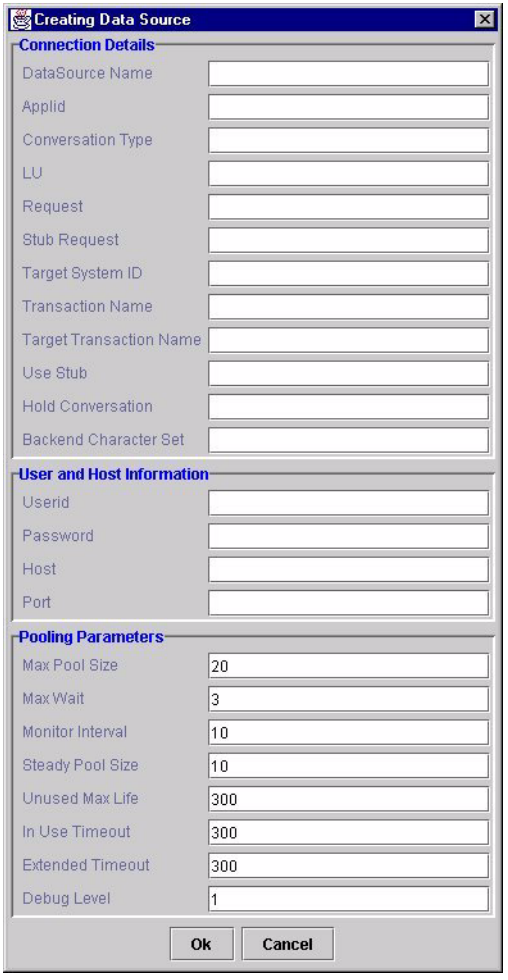
Creating a New Data Source

The data source contains all the information needed to connect to the CICS system, and stores all the function objects. In addition, the data source determines which system to mine, and where to place the function objects.

To Create a New Data Source

1. On the CICS Management Console, click the Create New Data Source icon. The Creating Data Source window is displayed. See Figure 4-8.

Figure 4-8 Creating Data Source Window



The Creating Data Source window contains information about the following:

- Connection Details
- User and Host Information

- Pooling Parameters

Connection Details

Table 4-2 lists information fields from the Connections Detail area needed to connect to the CICS system.

Table 4-2 Connection Details

Variable	Definition
DataSource Name	Name of the CICS backend system in the Repository.
Applid	This field specifies a CICS program name that is to be invoked on the CICS system.
Conversation Type	Can be either “POOLED”, or “DIRECT”. Pooled is used for TCP/IP and Direct is used for SNA/APPC.
LU	The Logic Unit (LU) name of the CICS in the mainframe. This is the logon APPLID of the CICS system. To find the value of the LU name: 1. Logon to the CICS system. 2. Type: cemt 3. Press Enter. 4. The following will appear at the bottom-right of the screen: APPLID = “XXXXXXXX” where XXXXXXXX is the LU name. The LU name is case sensitive and must be in upper case.

Table 4-2 Connection Details *(Continued)*

Request	Specifies the communication operation that the Relay (POOLED)/Client (DIRECT) performs. The following values are acceptable: SendAndEnd - Sends data to the server application, waits for a response, and then ends the conversation. At the end of the conversation the CICS SRVX transaction terminates. The socket remains open for additional conversations. If there is no additional conversation it closes after time-out. SendLast - Sends data to the server application and waits for a response. The CICS SRVX transaction waits for additional information from the client. If there is no additional information it closes after time-out. Note: SendLast will provide better performance than SendAndEnd. Send - Sends data to the server application. Receive - Receives data from the server application. End - Ends the conversation with the server. Quit - Aborts the conversation with the server.
Stub Request	Instructs SRVXLINK what to do. The following values are acceptable: EX - Execute the program without determining the unit of work. EC - Link to a program and commit to the unit at work. CO - Commit the unit of work. PR - The SRVXLINK is instructed to prepare to commit. RB - Rollback the unit of work.
Target System ID	Target CICS Region connected to the CICS data source.
Transaction Name	Informs CICS what program to start. In the CICS system, each transaction is identified by a four letter code, and is accessed with a program name.
Target Transaction Name	Used to specify the name of the transaction to be used only if TargetSRVX has been specified. If no name is specified, CICS supplies a default name.
Use Stub	Determines whether you are using the SRVXLINK program or not. Values are “Y” (Yes), or “N” (No).

Table 4-2 Connection Details *(Continued)*

Hold Conversation	Places the conversation on "hold" for the user until he wants to access it again. This option guarantees that the user will access the same CICS transaction on its next execution of an operation, within the same user interaction.
Backend Character Set	Enter the backend character set in the Backend Character Set Field in the Create or Edit Data Source dialog boxes. The connector converts the data from the client character set to the backend character set. Note, the client and backend character set must be compatible. In the servlet, call GX.setCharSet with the client character set. The PhoneBook sample supports different character sets.

User and Host Information

Table 4-3 contains information fields from the User and Host Information area that are specific to the user and host.

Table 4-3 User and Host Information

Variable	Definition
Userid	Used ID of the host ID.
Password	Password of the Mainframe host.
Host	Name of the host computer where CICS is running.
Port	The port where iPlanet Application Server is connected to the CICS system.

Pooling Parameters

Table 4-4 lists and defines parameters that describe the configuration of the CICS connection pool. The parameters are set by the system administrator. Default parameter settings appear.

Table 4-4 Pooling Parameters

Variable	Definition

Table 4-4 Pooling Parameters *(Continued)*

Max Pool Size	The maximum number of objects allowed in the pool. Set this number to the peak number of connections available to the server. If the number of objects is less than MaxPoolSize but exceeds SteadyPoolSize, objects are destroyed 3 seconds after they are returned to the pool.
Max Wait	The maximum time, in seconds, that a request for a physical object is held in the queue before the request times out and is destroyed.
Monitor Interval	(Optional) The time interval, in seconds, after which a thread is executed to monitor the current status of the pool. Default is 30 seconds. Typically, setting this number too low degrades performance, but it should be set to a number less than UnusedMaxLife.
Steady Pool Size	The number of unused objects that are kept in the pool until they time out.
Unused Max Life	The maximum time, in seconds, that a physical object remains unused in the pool. After this time, the physical object is destroyed.
In Use Timeout	Time interval set for an inactive "live" connection. When the set time is elapsed, the connection is closed.
Extended Timeout	Time interval set, in seconds, for an extended connection. When the set time is elapsed, the connection is closed.
Debug Level	(Optional) Determines type of message logging, as described by the following choices: 0: Logging turned off. 1: Logs only callback messages. 2: Logs all messages.

2. Enter in information into the fields, and Click OK.

   The new data source is created and an Information window displays informing you that must restart the Management Console to use the newly created data source. See Figure 4-9.

Figure 4-9 New Data source Information Message

3. Click OK, and restart the Management Console.

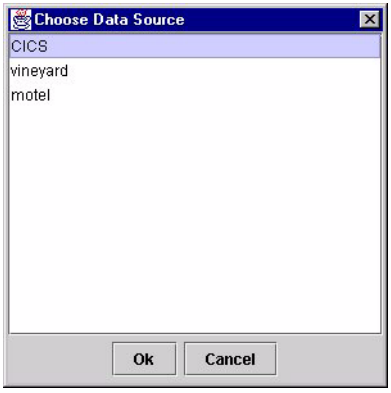
Editing a Data Source

You can edit an existing data source from the Management Console.

To Edit a Data Source

1. Click the Edit Data Source icon on the CICS Management Console to display the Choose Data Source list, see Figure 4-10.

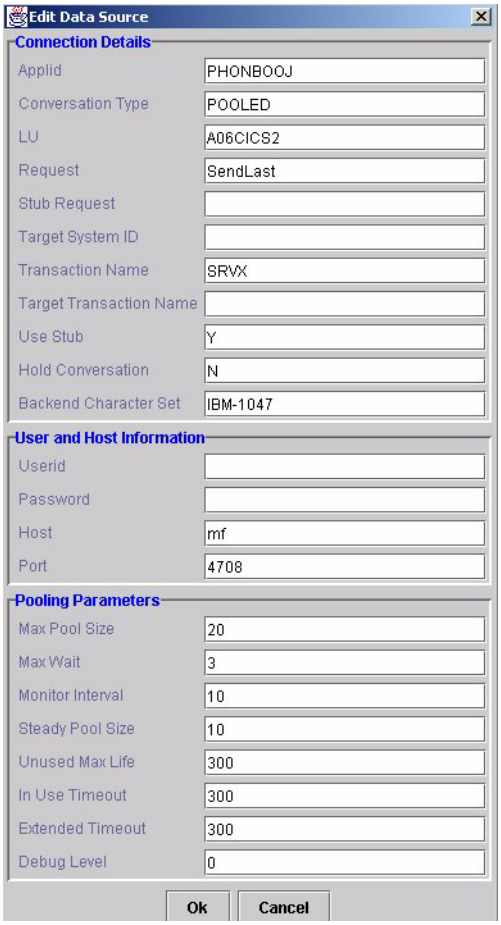
Figure 4-10 Choose Data Source



2. Select the data source to be edited, and click OK.

   The Edit Data Source window is displayed, see Figure 4-11.

Figure 4-11 Edit Data Source Window



3. Edit the appropriate fields and click OK.

   The data source is updated and loaded into the repository.

4. Restart the Management Console.

Mapping User Privileges

The CICS User Management function enables the administrator to assign access privileges to the backend system. There are two separate user identity domains: Web users and Enterprise System users. An incoming request has a specific web user identity associated with it. The web identity needs to be mapped to an Enterprise System user identity specific to the data source.

The user mapping service maintains user mapping tables. There are two distinct tables:

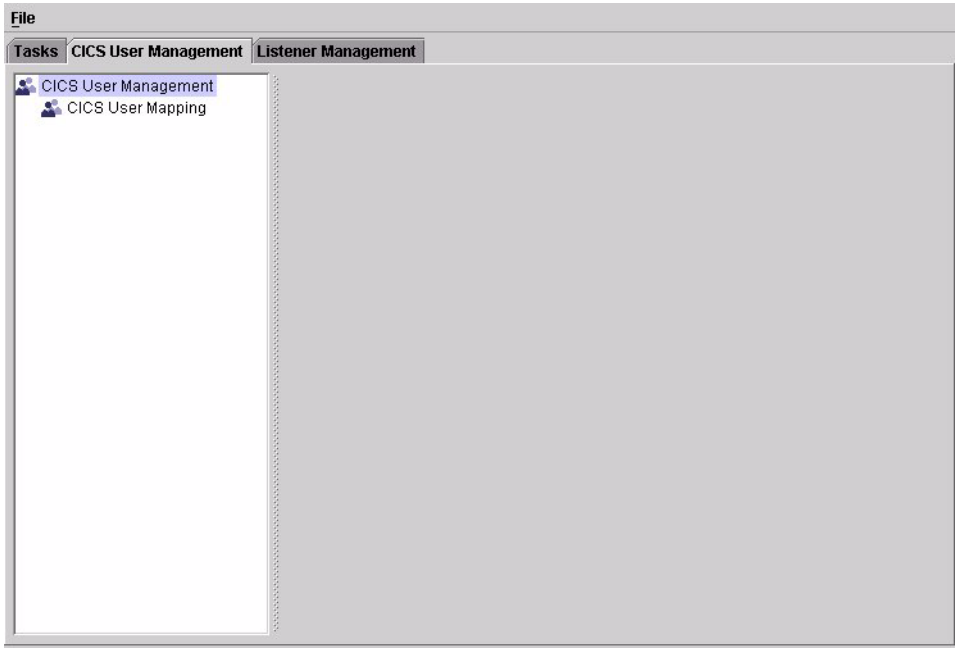
- Enterprise System user identities
- Web to Enterprise System mappings

The mapping service provides the connector with methods to determine the Enterprise System identity, given a web-identity, as per the mapping tables opened using the Management Console. The Enterprise System user table is populated using the Data Mining Tool in the Management Console. The data source administrator can populate the user mapping table interactively, through the Management Console.

To Perform User Mapping

1. From the CICS Management Console window, click the CICS User Management tab.
2. Click on the CICS User Mapping node. Figure 4-12 is displayed.

Figure 4-12 CICS User Management



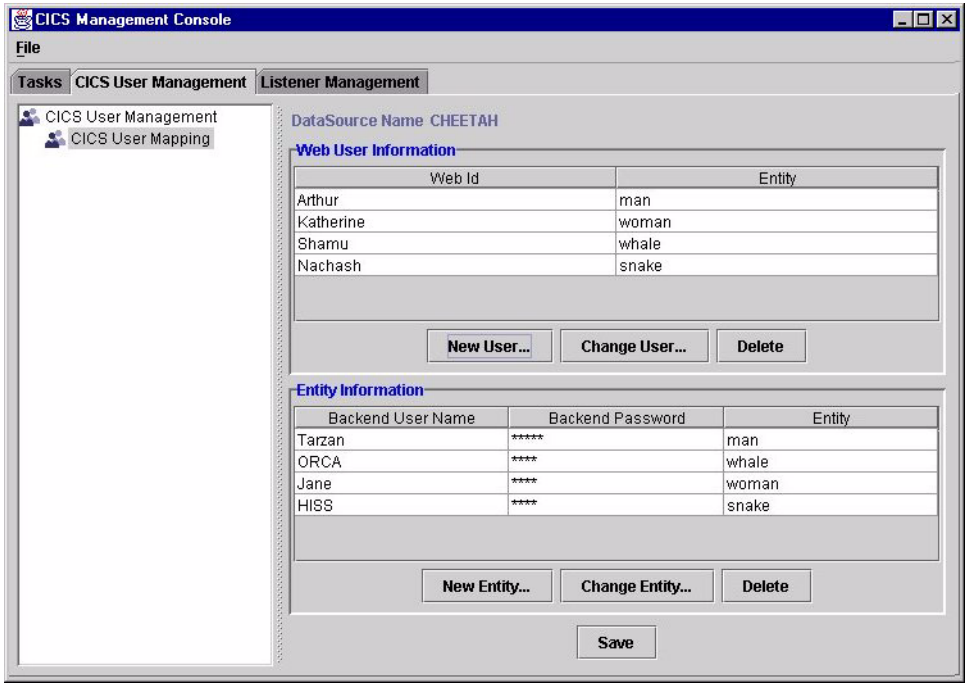
The tree view represents the general topics available. The browser view displays the details of the node selected.

3. Double click on the CICS User Mapping icon in the tree. A Getting Data Sources window (work in progress window) is displayed replaced by the Choose Data Source window. See Figure 4-10.

4. Select a data source from the Choose Data Source window and click OK.

   The CICS User Mapping window is displayed with the Web and Entity Information for the selected data source. See Figure 4-13.

Figure 4-13 CICS User Mapping



The CICS User Mapping window displays two fields of information:

- Web User Information
- Entity Information

Using the Web User Information area you can:

- Add New User
- Change User
- Delete a User

Using the Entity Information area you can:

- Add a New Entity
- Change Entity
- Delete an Entity

To save you changes click Save.

Web User Information

The Web User Information area is used to map a web-user identity to a back-end user identity. The Web ID is used by the server to enable users to move freely across multiple systems. The administrator assigns Entity Information before the Web User Information, since the Web User Information is based on the parameters defined in the Entity Information.

Table 4-5 lists and defines fields in the Web User Information area.

Table 4-5 Web User Information Parameters

Parameter Name	Definition
Web ID	ID assigned to users on the system.
Entity	Name of the assigned backend user. The naming convention is dependent upon the database type used.

Add New User

This option creates a new web user.

To Add a Web User

1. Click New User.

   The Web User Information window is displayed. See Figure 4-14.

Figure 4-14 Web User Information Window

2. Type in the Web ID.
3. Select an Entity Name from the available names.
4. Click OK when finished to return to the main window.

Change User

This option enables you to change an existing user.

To Change a Web User

1. Click Change User.
2. Edit the desired field.
3. Click OK. You are returned to the CICS User Mapping window.

Delete

This option enables you to delete an existing web user.

To Delete a Web User

1. Select the web user.
2. Click Delete.

   The web user is deleted and you are returned to the main window.

Entity Information

The Entity Information area is used to list data about back-end entities assigned to the web users. Table 4-6 lists and describes these parameters for the Entity Information window.

Table 4-6 Entity Information Parameters

Parameter Name	Definition
Backend User Name	Name of the backend user
Backend Password	Backend password, assigned by the user.
Entity	Backend entity name, assigned by the system administrator.

Add a New Entity

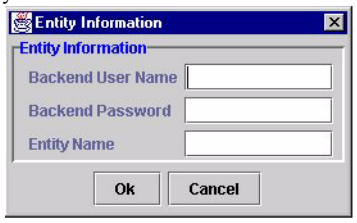
You can add a new backend entity in the CICS User Management.

To Add an Entity

1. To add a new backend entity, click New Entity.

   The Entity Information window is displayed. See Figure 4-15.

Figure 4-15 Entity Information Window



2. Enter the Entity information (Backend User Name, the Backend Password, and Entity Name).
3. Click OK.

   The new entity information is displayed in the Entity Information area of the window.

Change Entity

You can change the backend entity in the CICS User Management.

To Change Entity Information

1. Click Change Entity.
2. Edit the desired field(s).
3. Click OK.

   The edited entity is displayed in the Entity Information area of the window.

Delete an Entity

You can delete a backend entity.

To Delete an Entity

1. Select the entity.
2. Click Delete.

NOTE You can not delete a back-end entity if there are web users associated with the entity.

When you have completed all actions in the Entity Information window, click Save. All your additions and changes are saved to the system.

Listener Management

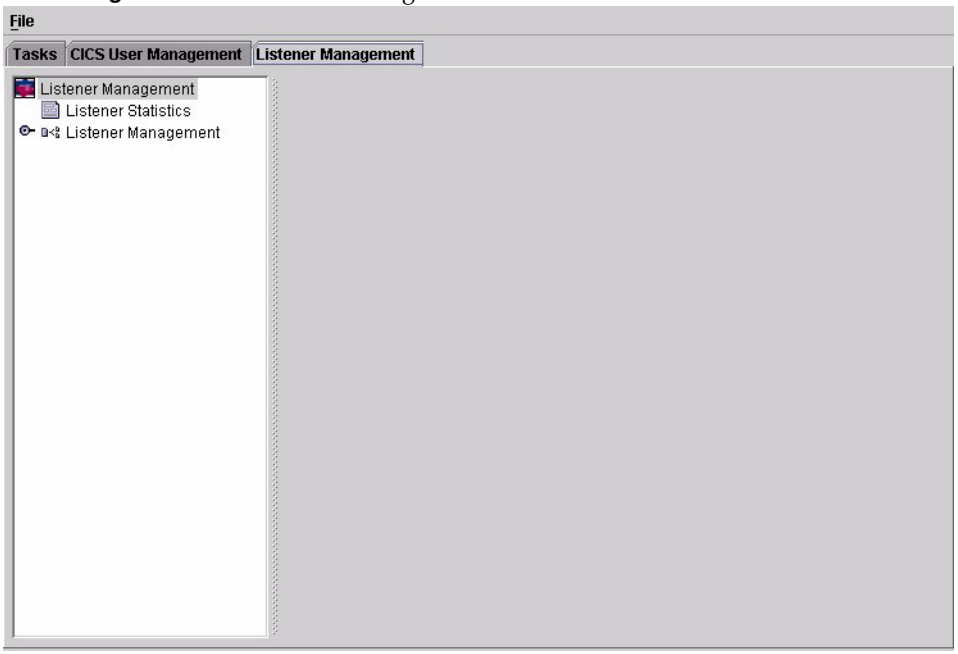
The Listener Management enables the administrator to view and edit Listener parameters, logon as a different user, and create new Listeners.

To Enter the Listener Management Function

1. From the CICS Management Console window, click the Listener Management tab.

   Figure 4-16 is displayed.

Figure 4-16 Listener Management Window



Listener Management has two subsections:

- Listener Statistics
- Listener Management

Listener Statistics

The Listener Statistics lists statistics associated with the Listener. See Figure 4-17.

Figure 4-17 Listener Statistics

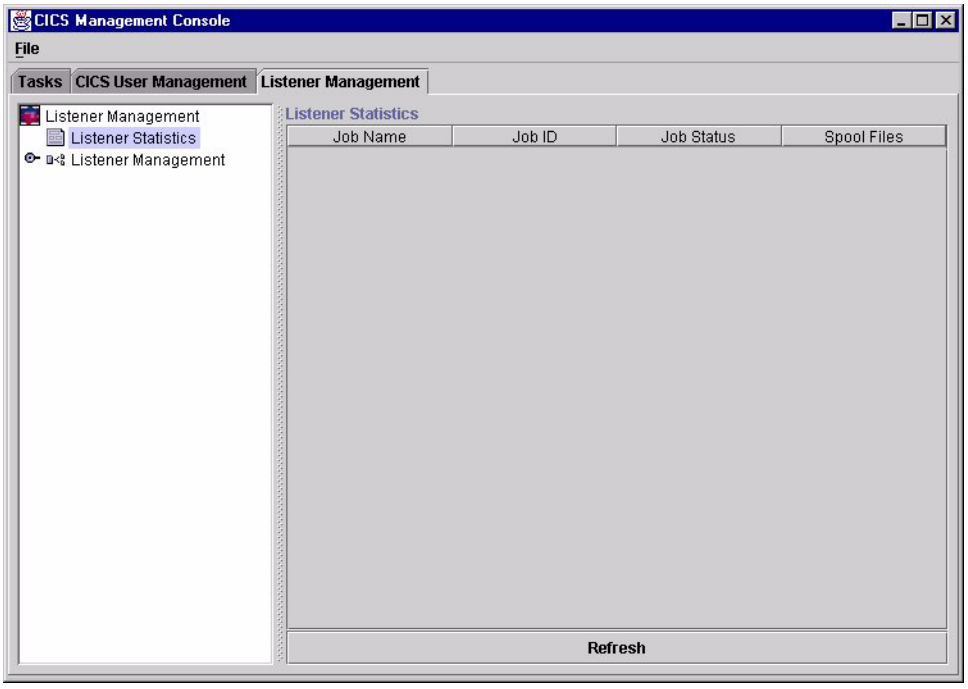


Table 4-7 lists and defines the variables of the Listener Statistics.

Table 4-7 Listener Statistics

Variable	Definition
Job Name	Name of the job under the Listener.
Job ID	ID of the job under the Listener.
Monitor Interval	Time period set for monitoring the Listener.
Spool Files	

To View Listener Statistics

1. In the Listener Management tab of the Management Console for CICS, click on Listener Statistics.

   The mainframe login data window is displayed. See Figure 4-18.

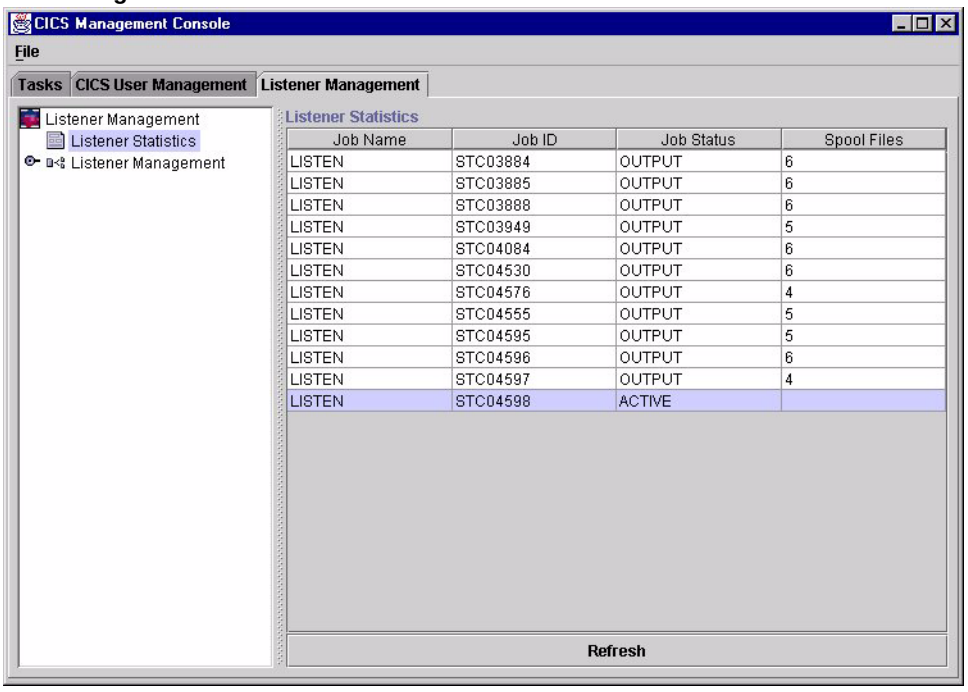
Figure 4-18 Mainframe Login Data

2. Type in the Login Name, Password, and Host Name for the mainframe, and click Login.

   The mainframe is logged in.

3. Click on Listener Statistics. The Listener Statistics are displayed, see Figure 4-19.

Figure 4-19 Listener Statistics



Refresh

To update the screen, click Refresh.

Listener Management

The Listener Management option allows you to:

- Login as a Different User
- Create (Start) New Listener

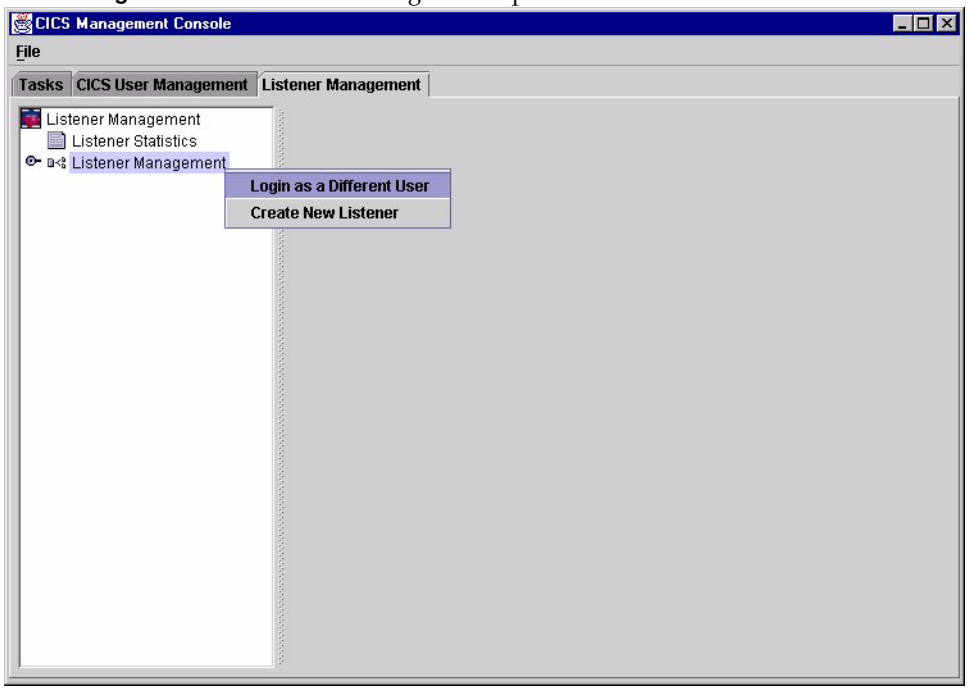
Login as a Different User

In the Listener Management window, you can login as a different user to view information about the Listener, or create a new Listener.

To Login as a Different User

1. In the Listener Management tab, right mouse click on the Listener Management option. A pop up menu is displayed, see Figure 4-20.

Figure 4-20 Listener Management Options



2. Select the Login as a Different User option.

   The Mainframe Login Data window is displayed. See Figure 4-21.

Figure 4-21 Mainframe Login Data Window

Table 4-8 describes the Mainframe Login Data fields.

Table 4-8 Mainframe Login Data

Variable	Definition
Login Name	Login name used to login to the mainframe.

Table 4-8 Mainframe Login Data *(Continued)*

Password	Password used for accessing the mainframe.
Host Name	Name of the host where the mainframe is located.

3. Enter in the mainframe login data, and click Login.

   You are now logged in as a different user.

Create (Start) New Listener

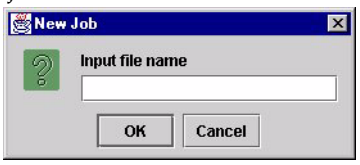
Listeners are created in the mainframe computer. This option allows you to start a new listener on a port other than the one already used.

To Create New Listener

1. In the Listener Management tab, right mouse click on the Listener Management option. A pop up menu is displayed, see Figure 4-20.
2. Select the Create New Listener option.

   The New Job window is displayed. See Figure 4-22.

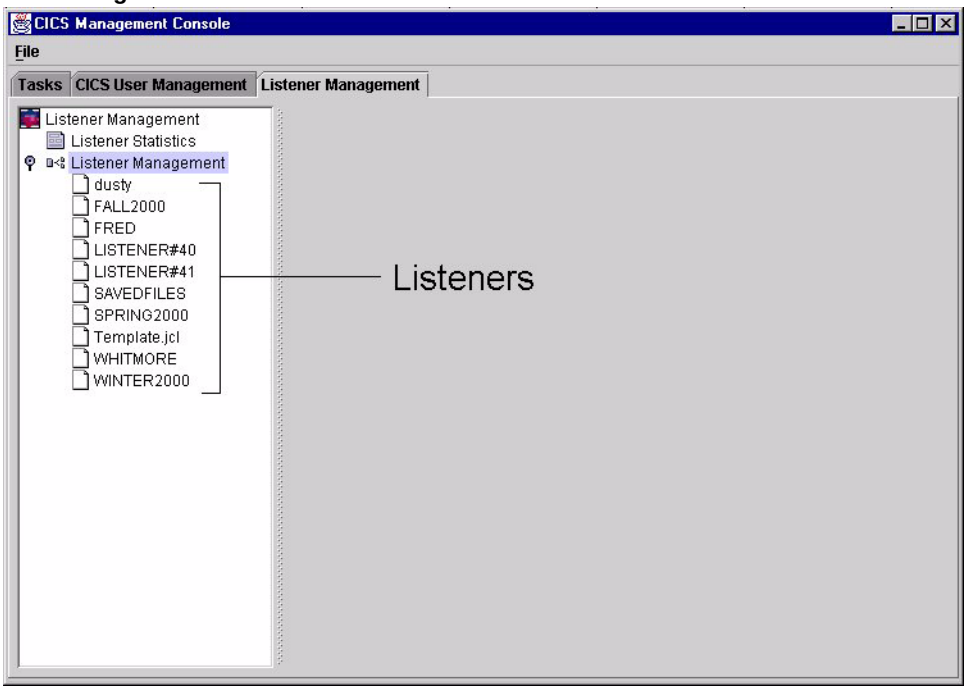
Figure 4-22 New Job Window



3. Type in a new file name, and click OK.

   A new Listener is created.
4. You can check the new Listener listing by double clicking on the Listener Management option. The Listeners are displayed in a tree, see Figure 4-23.

Figure 4-23 Listeners



The Repository Browser

All the definitions for the connector using the Management Console are saved in a repository. The Repository Browser is the Graphical User Interface to the repository and is used only for viewing information about the connector. Most management tasks are done using the Management Console.

The Repository Browser contains the following:

- information representing the connection to the back-end system
- all the function objects that you mine from the back-end system
- user mapping information
- connection pools

All of these components are part of a data source.

Although the repository service does not impose any specific organization (schematics) on the repository contents, the UIF runtime requires the contents of the repository to be organized in a specific and well-defined way. This organization is the UIF repository schematics. The UIF runtime expects specific information in specific places in the repository hierarchy. Therefore, repository contents should not be arbitrarily modified by the user. Contents are modified as part of the specific administration activities controlled by the Management Console, which enables the administrator to use the import and export functions appropriately. In addition, the UIF runtime service runs a consistency check on repository contents at startup.

Repository Browser Functions

The following functions are available in the Repository Browser:

- Import — allows you import XML files into the repository
- Export — allows you export XML files from the repository
- Delete — deletes nodes from the data structure in the repository
- Refresh — refreshes the Repository Browser display
- Import Root — imports an XML file and creates the root node

Loading Data Into the Repository Browser

There are two ways to load data into the Repository Browser. You can either load the data automatically into the Repository Browser via the Management Console, or you can manually load the XML file using the Import function.

To Access the Repository Browser from Windows NT/2000

- Select Start > Programs> iPlanet Application Server 6.5> UIF 6.5 Repository Browser.

  The Repository Browser is displayed. See Figure 4-24.

To Access the Repository Browser from Solaris

1. Enter the following command lines:

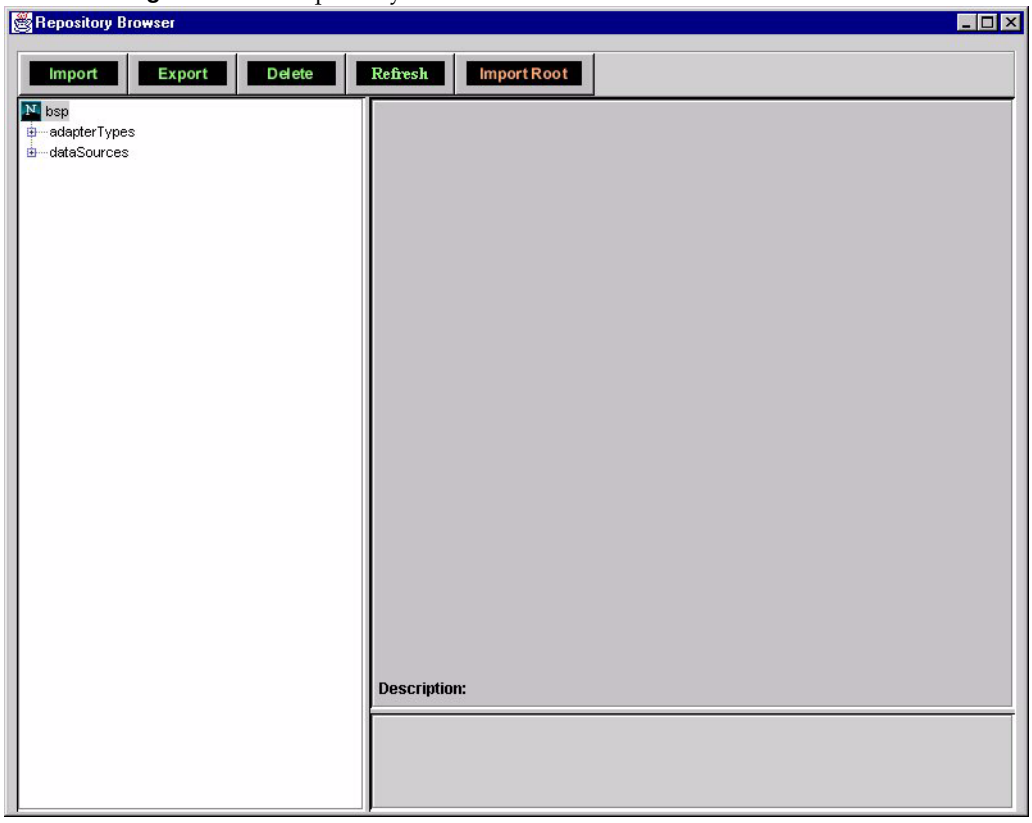
   `cd iplanet/ias6/ias/APPS/bin`

   and

`./bspbrowser.sh`

   The Repository Browser is displayed. See Figure 4-24.

Figure 4-24 Repository Browser



The Repository Browser is divided into two panes. When you open the browser the left pane displays nodes containing the adapter (connector) types and data sources. These nodes are hierarchical and can be expanded to show details of the data structure and function objects. The right pane displays the properties of the node selected in the left pane.

For a more detailed description of the node contents, refer to the *iAS Enterprise Connector for CICS Developer's Guide*.

The Repository Data Structure

The Repository Browser contains the following nodes that are installed automatically when you install the UIF:

- root node
- adapterTypes
- dataSources

adapterTypes

The adapterTypes node contains a default data structure for each type of connector that is installed. The adapterTypes section contains one subtree entry per adapter. The entry contains common definitions, which form the basis for defining data sources on the connector.

dataSources

The dataSources node contains a default dataSource structure that is set when you install a connector. You can define several data sources for each adapterType. The dataSource section contains subtree entries, each of which is a logical UIF data source. A data source corresponds to a specific back-end Enterprise System and is supported by the corresponding adapterType.

The data source definition activity is tightly controlled by the Management Console. There are two distinct user roles: the data source administrator and the application developer. The data source administrator is responsible for creating and administering data source entries, while the application developer develops applications that access these data sources.

A data source definition is usually heavily pre configured by the data source administrator. The application developer usually uses pre configured function objects, much like EJBs would be used with the deployment descriptor already set. This substantially isolates the application developer from the back-end specific interaction issues. See the *iAS Enterprise Connector for CICS Developer's Guide* for more information about programming applications.

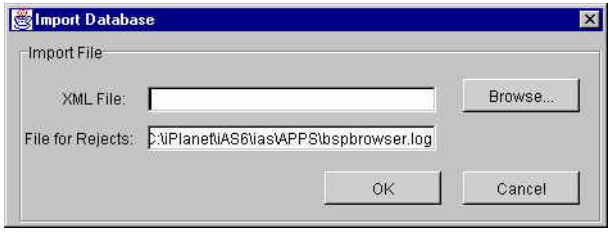
Import

To Import XML Files

1. Select Import.

   Figure 4-25 is displayed.

Figure 4-25 Import Database Window



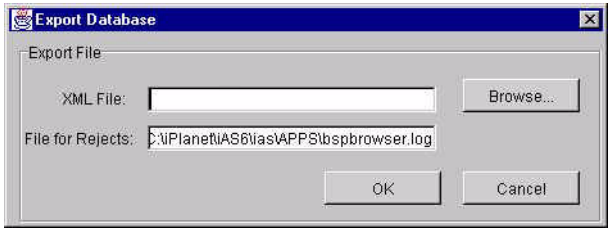
2. Enter the XML filename that you want to import and click OK.

To Export XML Files

1. Select Export.

   Figure 4-26 is displayed.

Figure 4-26 Export DataBase Window



2. Enter the XML filename that you want to Export and click OK.

To Delete Nodes

1. Select Delete.
2. Select the Node you want to delete

   A warning message is displayed.
3. Click OK.

   The display in the Repository Browser is updated with current data.

To Refresh the Display of Repository Contents

- Click Refresh.

To Import the Root Node into the Repository Contents

1. Click Import Root.

   The Import Database window is displayed as shown in Figure 4-25.
2. Enter the XML filename to import and click OK.

   The root node and its contents are imported into the repository.

Chapter 5

Configuring the Application

This chapter is intended for administrators who configure operational parameters that support the UIF for the iPlanet Application Server Enterprise Connector for CICS. Most configuration parameters are stored in the UIF repository. The CICS Management Console enables you to modify these parameters.

This chapter describes pooling concepts when you set up a pooling configuration node in the UIF repository.

The chapter contains information about the following topics:

- About Pooling
- Bind Duration
- Pooling Configuration

About Pooling

Pooling enables you to share resources. In the UIF, pools are used to share connections to a backend system from the iPlanet Application Server. Service providers allocate and reuse connections from pools to avoid unnecessary creation and destruction of connections to the backend. Pools are defined in the Repository on a per-data-source basis. This means that each data source defines its own connection pool, which can be individually configured.

An object in a UIF pool represents a connection to a backend system. Each pool contains a specified maximum number of objects to constrain the maximum number of simultaneous connections. A pool typically holds a number of objects that represent the number of connections in a steady state.

When a iPlanet Application Server application enables a service provider, the connector attempts to obtain a connection object from the pool.

If the maximum number has been reached and no object is available after a specified waiting period, the request times out. You can configure the time out period.

If a pool object remains unused for a period of time, the object is destroyed. You can specify how long to wait before the object is destroyed.

The use and longevity of pool objects are monitored by a thread that executes periodically. You can specify the interval after which the monitor thread executes. Typically, the internal thread execution is smaller than the waiting period before an object is destroyed. However, setting the monitoring interval too small can degrade performance. You specify the frequency in the `MonitorInterval` node of the pooling connection; for more information see Pooling Configuration.

Bind Duration

A pooled connection is bound to a servlet for a specified period of time, called a bind duration, which is specified in a service provider type. Typically, a bind duration is only for the length of time it takes to execute the method requiring the connection, although some operations, such as beginning a transaction, may require a longer bind duration.

UIF supports three bind durations. Table 5-1 lists and describes these supported blind durations.

Table 5-1 Bind Durations

Bind Duration	Description
method-bound-poolable	The pooled connection is bound for the duration of the method call.
sp-bound-poolable	The pooled connection is bound for the life of the service provider; the connection is then returned to the pool.
not-poolable	The connection is not poolable. The connection is bound for the life of the service provider and is destroyed when the service provider is destroyed.

Bind durations are specified in the repository. The following example shows that the <adaptor>Conn service provider type's bind duration is method-bound-poolable:

```
serviceProviderTypes

<adaptor>Conn
```

```
bindDuration string method-bound-poolable
```

Bind Duration Escalation

The bind duration may be changed dynamically by the connector. For example, at the beginning of a transaction, a connector may escalate the bind duration from method-bound to sp-bound, and then reset the bind duration to method-bound at the end of the transaction.

Bind Duration Time-outs

An sp-bound connection cannot be used by another service provider until the connection has been released by the first service provider. If the first service provider does not release the connection in a timely manner, a time-out occurs and the connection is forcibly released and returned to the pool. The time-out prevents a "runaway" service provider that failed to call disable() from keeping the connection indefinitely.

Pooling Configuration

You can specify the pooling configuration for a service provider by specifying the pooling configuration characteristics in the repository. Table 5-2 lists and describes pooling configuration parameters.

Table 5-2 Pooling Configuration Parameters

Parameter	Description
MaxPoolSize	The maximum number of objects allowed in the pool. Set this number to the peak number of connections available to the server. If the number of objects is less than MaxPoolSize but exceeds SteadyPoolSize, objects are destroyed three seconds after they are returned to the pool.
SteadyPoolSize	The number of unused objects that are kept in the pool until they time out. Set SteadyPoolSize to the steady state number of connections available from the host server.
MaxWait	The maximum time, in seconds, that a request for a physical object is held in the queue before the request times out and is destroyed.

Table 5-2 Pooling Configuration Parameters *(Continued)*

UnusedMaxLife	The maximum time, in seconds, that a physical object remains unused in the pool. After this time, the physical object is destroyed.
UnusedMaxLife	(Optional) The time interval, in seconds, after which a thread is executed to monitor the current status of the pool. Default is 30 seconds. Typically, setting this number too low degrades performance, but it should be set to a number less than UnusedMaxLife.
DebugLevel	(Optional) Determines type of message logging, as described by the following choices: 0: Logging turned off. 1: Logs only callback messages. 2: Logs all messages.

The example below is used to show a "generic" pool, for specific information about the connector. The following entries define the pool configuration for the <adapter>Conn service provider:

Code Example 5-1 Pool Configuration

```
pools
         <adapter>Pool
            sptype reference

bsp.adapterTypes.<adapter>.serviceProviderTypes.cicsConn
            config:do
               DebugLevel integer 1
               MaxPoolSize integer 20
               MaxWait integer 3
               MonitorInterval integer 10
               SteadyPoolSize integer 10
               UnusedMaxLife integer 300
```

Appendix A

Operating the TELCO Sample

This appendix describes how to configure, install and operate the Telco Customer Service sample that requires DB2 for OS390. The sample demonstrates connector capabilities for CICS and DB2.

The following topics are described:

- Uploading Files for the TELCO Sample
- Installing the TELCO Sample on the Mainframe
- Activating the TELCO Sample

Uploading Files for the TELCO Sample

You willonly need to upload files before installing the TELCO sample .if you did not already upload these files during the mainframe installation. See Chapter 3, “Installing the Mainframe Component”.

To Upload Files for the TELCO Sample

1. Allocate one data set on MVS with `1recl=80, blksize=3120, recfm=fb.`

   For example, `IAS.CICS.DBRM.XMIT`.

2. Use the 3270 emulation program to transfer the file DBRM.XMT to the respective MVS data set.

   Use the MVS/TSO binary transfer type and fixed-file option.

3. Issue the following TSO command:

   ```
   RECEIVE INDATASET(IAS.CICS.DBRM.XMIT).
   ```

4. At the prompt, enter the following:

```
DATASET(IAS.V6R5M0.DBRM.LIB)
```

   where `IAS.V6R5M0.DBRM.LIB` is the name of the Sample-2 (TELCO) DBRM library.

Installing the TELCO Sample on the Mainframe

The TELCO sample should be installed to verify that CICS connector is installed properly. Be certain to adapt the JCL jobs to your installation requirements.

To Install the TELCO Sample

1. Adapt and run the CRETAB member in the CICS installation library to define DB2 objects.
2. Adapt and run the BIND member in the CICS installation library to bind packages for TELCO sample programs.

   This step requires the supplied DBRM library.
3. Adapt and run the CICSDEF2 member in the CICS installation library to define sample TELCO programs in the CICS Resource Definition.

   This step also contains the definition for the CICS-DB2 attachment.
4. Ensure that the supplied CICS LOAD library containing the TELCO sample modules is present in the DFHRPL list during the CICS startup.
5. Restart CICS and the CICS-DDB2 attachment after completing all of the installation steps.

Activating the TELCO Sample

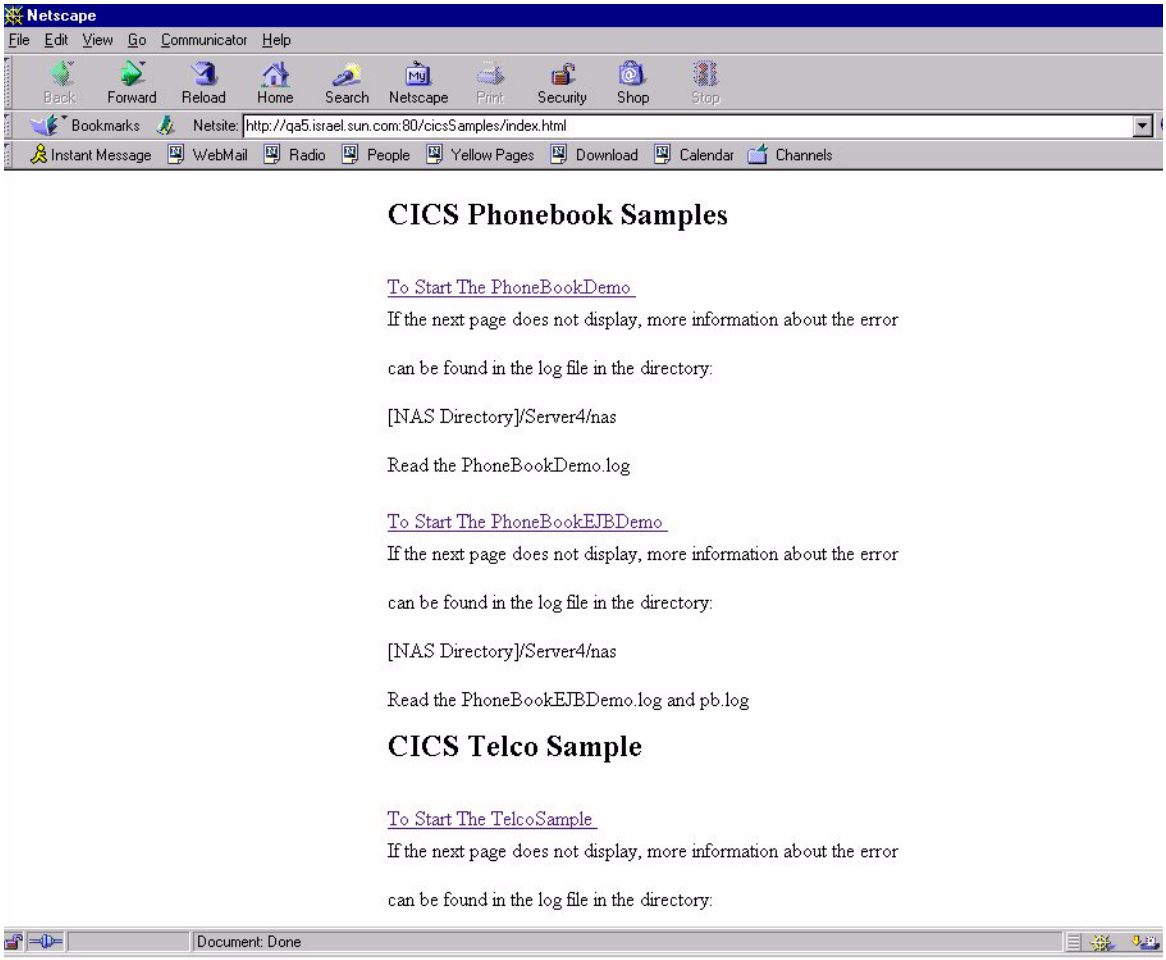
After you have uploaded the files and installed the TELCO sample software you can activate the sample and verify that the CICS connector is installed properly.

To Activate the Samples

1. Enter the following URL:

   http://localhost/cicsSamples/index.html

Figure A-1 CICS Phonebook Sample

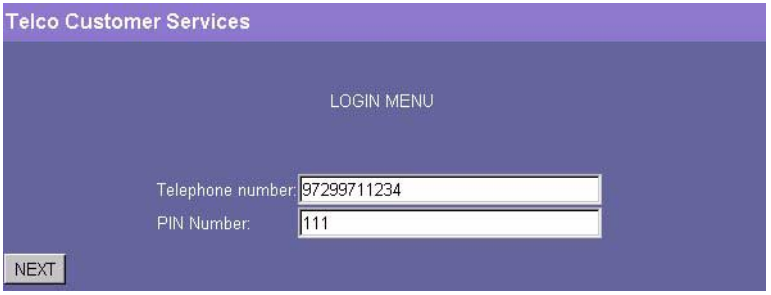


2. Click on “To Start the Telco Sample”.

To Log In

1. Type your telephone number and PIN number in the Login Menu dialog box, as seen in Figure A-2.

Figure A-2 Login Menu



2. Click Next to display the Telco Customer Service Main Menu. See Figure A-3.

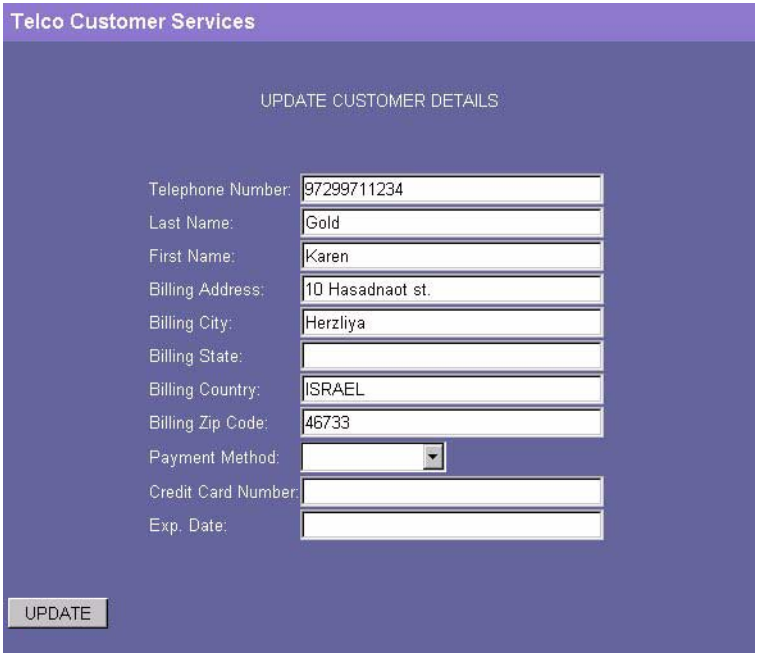
Figure A-3 Main Menu

To Update Customer Details

- Click Update Customer Details.

  The Update Customer Details dialog box appears. See Figure A-4. This dialog box is used to update the name, address, and telephone number of the customer. The payment method, credit card number, and credit card expiration date are not displayed and cannot be changed.

Figure A-4 Update Customers Details



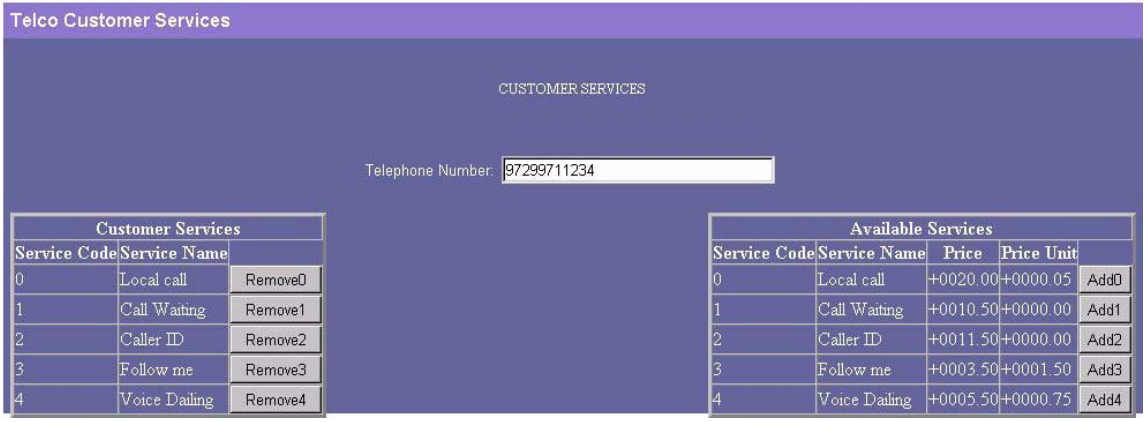
3. Enter the changes and click Update.
4. Press Enter to refresh the main menu.
5. Press the Back button to return to the main menu.

To Display Customer Services

- From the main menu click Services.

  The Customer Services dialog box appears. See Figure A-5.

Figure A-5 Customer Services



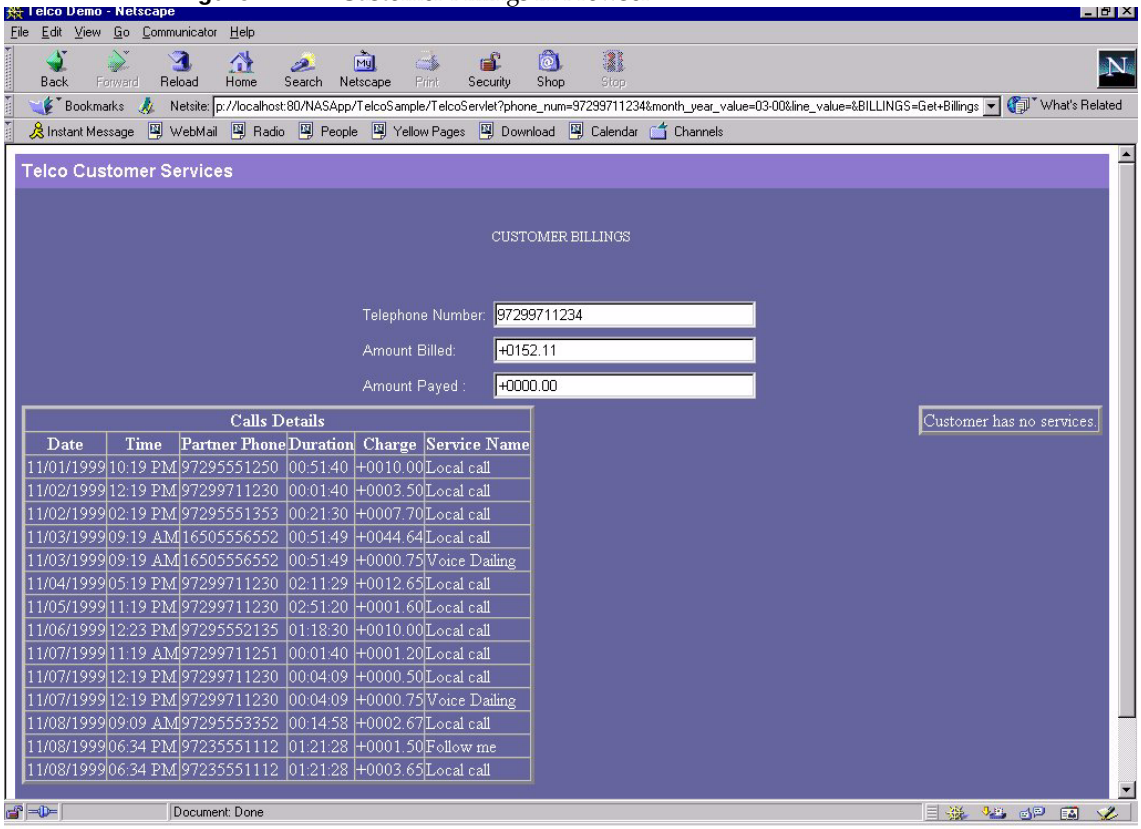
To Display the Customer's Monthly Billing

1. Enter the telephone number and billing month, and click Get Billings. See Figure A-6.

Figure A-6 Customer Billings Window

The Customer Billings window is displayed, as seen in Figure A-7.

Figure A-7 Customer Billings in Browser



Security Exit for Mainframe Listener

The Mainframe Listener has a security exit.

The exit:

- is called immediately after the first message is received from the application server
- can be modified to the customers requirements
- must be written in Assembler language

The name of the module must be SRVSCURE and must reside in an APF authorized library in the listeners STEPLIB, preferably in the regular listener load library.

It is highly recommended to write the exit as re-entrant.

The default exit does not perform any security checks.

On entry to the exit, *register 1* points to the following structure:

When you complete the exit, set R15 to zero for normal processing, anything else terminates the incoming request.

Input

The structure that R1 points to on entry is shown in the following code:

Code Example B-1 Code where R1 Points

```
SCUR_IP   DS   CL4   CLIENT IP ADDRESS
SCUR_PORT DS   CL2   LISTENER PORT
          DS   CL2   FILLER
SCUR_LIST DS   CL8   LISTENER STARTED TASK NAME
          DS   CL9   FILLER
```

Code Example B-1 Code where R1 Points

```
SCURUSER  DS   CL16   SAF USERID
SCURPASS  DS   CL16   SAF PASSWORD
SCURCONN  DS   F      POINTER TO CONNECT RECORD HDR
SCURDATA  DS   F      POINTER TO DATA AREA IN CONNREC
SCURRET   DS   C
```

Table B-1 Security Exit Field Definitions

Field	Description
SCUR_IP	The 4 byte IP address of the client machine in binary format. For example, if the client IP address is 129.156.62.108 then SCUR_IP will have the value of : x'819C3E6C'
SCUR_PORT	The port number that the listener is listening on.
SCUR_LIST	The name of the listener started task.
SCURUSER	The userid supplied by the incoming request.
SCURPASS	The password supplied by the incoming userid
SCRUDATA	A pointer to the data in the message.

Output

Set R15 to zero for normal processing, anything else terminates the incoming request.

Samples

We provide four members in the Relay installation library:

Table B-2

Member Name	Description
SRVSCURE	This member provides the source of the default exit as supplied in the load library
SECUREX1	This member provides a sample exit that verifies if a valid userid and password have been provided.

Table B-2

Member Name	Description
SECUREX2	This member provides a sample exit that verifies if the provided userid has read access to a specific facility.
ASMSCURE	This member provides JCL to assemble and link a Security exit.

This exit can be used to perform any kind of security checks. You can for example call RACROUT and verify that the user has access to a particular resource or check the IP address of the incoming request against a list.

Example - SECUREX2

The characteristics of the `SECUREX2` sample are:

- re-entrant
- performs a RACROUT verify if the user has read-access to a facility called 'IPLANET'. If the user is authorized, normal processing continues, otherwise it fails.

ThE following is the sample SECUREX2:

Code Example B-2 SECUREX2 Sample Code

```
SRVSCURE CSECT
SRVSCURE RMODE ANY
SRVSCURE AMODE 31
*****************************************************************
* HOUSEKEEPING FOR RE-ENTRANT CODE
         USING *,R3                    CODE ADDRESSABILITY
         STM   R14,R12,12(R13)         SAVE REGISTERS
         LR    R3,R15                  SET UP ENTRY POINT REGISTER
         LR    R12,R13                   SWITCH SAVE AREA
         LR    R4,R1                   SAVE POINTER TO PARAMETER
* OBTAIN STORAGE FOR DYNAMIC SAVE AREA
         STORAGE OBTAIN,LENGTH=4*18,ADDR=(R1)
         LR    13,1                    R13 -> SAVEAREA
         ST    12,4(13)                LINK CALLERS SAVE AREA
         ST    13,8(12)                LINK OUR SAVE AREA IN CALLERS
* END OF HOUSEKEEPING
         USING SECURE,R4                  ADDRESS PARAMETER BLOCK
         MVI   SCURRET,X'00'              ASSUME GOOD RETURN
*GETMAIN FOR DYNAMIC WORKAREA
         STORAGE OBTAIN,LENGTH=WORK_LEN,ADDR=(R5)
         USING WORK,R5              ADDRESS WORK AREA
```

Code Example B-2 SECUREX2 Sample Code

```
         LA    R8,RACD              POINT TO DYNAMIC LIST
         LA    R10,RACS             POINT TO STATIC  LIST
         L     R9,RACLEN            LENGTH OF LIST
         LR    R11,R9
         MVCL  R8,R10               MOVE IT IN
         MVC   IP_PROF,=CL39'IPLANET'  FACILITY NAME
         LA    R6,IP_PROF           R6 -> FACILITY NAME
         LA    R7,RACWK             R7-> RACF WORK AREA
         MVC   USER,SCURUSER         MOVE IN USERID FROM CLIENT
         LA    R9,USER              AND POINT TO IT
*ISSUE RACROUTE TO SEE IF USER HAS READ AUTHORITY TO FACLITY IPLANET
         RACROUTE REQUEST=AUTH,ENTITY=((R6)),WORKA=(R7),              X
               USERID=(R9),MF=(E,RACD),RELEASE=2.6
         LTR   R6,R15      Q. USER AUTHORIZED?(AND SAVE RETCODE IN R6)
         BZ    EXIT          Y. EXIT WITH GOOD RETURN CODE
         MVI   SCURRET,X'15' N. INDICATE BAD RETURN
EXIT     DS    0H
         STORAGE RELEASE,LENGTH=WORK_LEN,ADDR=(R5)
* END OF PROGRAM HOUSKEEPING
         LR    R1,R13       KEEP SAVE AREA ADDRESS
         L     R13,4(,R13)  R13 - > CALLER SAVE AREA
         STORAGE RELEASE,LENGTH=4*18,ADDR=(1) RELEASE SAVE AREA
         L     R14,12(,R13)    GET RETURN ADDRESS
         LR    R15,R6          SET RETURN CODE
         LM    0,R12,20(R13)   RESTORE CALLERS REGISTER
         BSM   0,14            AND RETURN
**  CONSTANTS
RACS     RACROUTE MF=L,REQUEST=AUTH,CLASS='FACILITY',                  X
               ATTR=READ,RELEASE=2.6
RACLEN   DC   A(*-RACS)
**  DSECTS
WORK     DSECT
RACD     RACROUTE MF=L,REQUEST=AUTH,CLASS='FACILITY',                  X
               ATTR=READ,RELEASE=2.6
RACWK    DS   128F
IP_PROF  DS    CL39
USER     DS   CL8
CLASS    DS   CL8
WORK_LEN EQU *-WORK
SECURE   DSECT
SCUR_IP   DS   CL4   CLIENT IP ADDRESS
SCUR_PORT DS   CL2   LISTENER PORT
          DS   CL2   FILLER
SCUR_LIST DS   CL8   LISTENER STARTED TASK NAME
          DS   CL9   FILLER
SCURUSER DS    CL16   SAF USERID
SCURPASS DS    CL16   SAF PASSWORD
SCURCONN DS    F      POINTER TO CONNECT RECORD HDR
SCURDATA DS    F      POINTER TO DATA AREA IN CONNREC
         DS    F      FILLER
SCURRET  DS    C      RETURN CODE
**  REGISTER EQUATES
R0    EQU 0
R1    EQU 1
```

Code Example B-2 SECUREX2 Sample Code

```
R2    EQU 2
R3    EQU 3
R4    EQU 4
R5    EQU 5
R6    EQU 6
R7    EQU 7
R8    EQU 8
R9    EQU 9
R10   EQU 10
R11   EQU 11
R12   EQU 12
R13   EQU 13
R14   EQU 14
R15   EQU 15
         END
```

Compile and Link JCL Code

The exit can be compiled and linked using the following JCL. The JCL code is supplied in the Relay install library member: ASMSCURE:

Code Example B-3 JCL Code for Compile and Link:

```
// JOB
//   EXEC ASMACL,
//  PARM.C='NODECK,OBJECT,USING(MAP,WARN(11)),RENT',
//  PARM.L='LIST,LET,AMODE=31,RMODE=ANY,RENT,REUS,REFR'
//SYSIN DD DSN=iplanet.SOURCE(SRVSCURE),DISP=SHR
//L.SYSLMOD DD DSN=iplanet.LOAD(SRVSCURE),DISP=SHR
```

Glossary

Application Programming Interface (API) The interface to a library of language-specific subroutines that implement higher level functions. A set of calling conventions defining how a service is invoked through a software package.

Applet A component that typically executes in a web browser, but can execute in a variety of other applications or devices that support the applet programming model.

Applications Programmer Responsible for writing servlets or EJBs that call the UIF API. Uses the Repository Browser to determine the available data types and access methods.

Array Object Contains data objects or primitive values as elements in the object. Array elements must be homogeneous. Each element within the array object is referred to by an integer that specifies its position in the array object.

Attribute Field Attributes that describe allowable attributes for the field where the input and output are located.

CICS (Customer Information Control System) An IBM communications program designed to allow transactions entered at a remote site to be processed concurrently by a mainframe host.

Daemon A program that is not explicitly invoked, and remains idle until summoned (called on).

Data Block Describes the input and output of operations. The data block can only contain two structures: input and output. All input and output structures contain fields that can be only one of the following types: primitive, structure, or array.

Data Object Used by the UIF to represent data or metadata in a generic fashion. Data objects are used to exchange data between a servlet and the UIF, and between the UIF and the connector.

Data Source Contains all the information needed to connect to the PeopleSoft system, and stores all the function objects. In addition, the Data Source determines which system to mine, and where to place the function objects.

Deployment Deploying an application includes installing all of the application's files, and registering all of its components on the destination server. You deploy an application using the Deployment Tool, a separate tool accessible from the iPlanet Application Server (iAS). An application must be deployed before it can be used.

EJB (Enterprise Java Beans) A server-side component architecture for writing reusable business logic and portable enterprise applications. They are written entirely in Java and run on any EJB compliant server. They are operating system, platform, and middleware independent, thereby preventing vendor lock-in.

EIS (Enterprise Information System) Referred to as a backend system.

Enterprise Connector The component in iPlanet Application Server Enterprise Connector for R/3, PeopleSoft, Tuxedo, or CICS that enables you to access the appropriate backend system.

ERP (Enterprise Resource Planning) A multi-module software system that supports enterprise resource planning. An ERP system typically includes a relational database and applications for managing purchasing, inventory, personnel, customer service, shipping, financial planning, and other important aspects of the business.

Function Object A group of business methods available for execution on the specific enterprise server. These objects are derived from metadata mined from the enterprise server that share a common state.

iAS (iPlanet Application Server) The iAS provides the most robust e-commerce platform for delivering innovative and leading-edge application services to a broad range of servers, clients, and devices.

iWS (iPlanet Web Server) A web server that is ideally suited to the Java development community for use as the development and test platform for web applications.

Java An object-oriented programming language developed by Sun Microsystems, Inc. to create executable content (i.e, self-running applications) that can be easily distributed through networks like the Internet.

Load Balancing Load Balancing is the configuration of a computer system, network, or disk subsystem to more evenly distribute the data and/or processing across available resources in order to increase the speed and reliability of transmissions.

Operations Directory A directory with operations that contain data blocks and property sets.

Primitive Object A data type that contains a single value of an integer, float, double, fixed-length string, or variable-length string.

Repository A specialized structure where all the module's functions are stored for the use of the iPlanet Application Server Enterprise Connector.

Repository Browser The component that enables you to browse data (content) in the repository, and to view the available functions (input and output parameters) for the backend system.

Runtime Object The entry point into the UIF.

Service Provider Object The logical representation of a connection to a back-end system, which must be enabled before it can be used. Typically, the service provider object is not bound to a physical connection until absolutely necessary.

Server Tier The server tier is represented by an application server and optionally a web server such as the iPlanet Web Server Enterprise Edition. The server tier houses the business logic (Enterprise Java Beans of your application servlets), and provides scalability, high availability load balancing, and integration with a variety of data sources.

Servlet An applet that runs on a server, usually meaning a Java applet that runs on a Web server.

Structure Object Contains other data objects or primitive values whose fields are heterogeneous such as fields, and whose fields are heterogeneous. Each object within the structure object is referred to by a string that represents the field name. Field names have a maximum length of 32 characters.

System Name The system name used. For load-balancing connection only.

Three-tier Application Model A model of an application system that is composed of the following three tiers: Client, Server, and Backend (EIS).

Type Information Objects Structured objects that contain the type information of a data object; i.e. definition of the fields in a structure and the fields corresponding data types. Instances of data objects can be created of type information objects. Each of these instances contain a reference to a type of information object. Numerous data types can share the same type information object.

UIF (Unified Integration Framework) An application programming framework that provides a single Application Programming Interface (API) to access different backend systems.

URL (Universal Resource Locator) An address for a resource or site (usually a directory or file) on the World Wide Web, and the convention that web browsers use for locating files and other remote services.

XML (eXtensible Markup Language) A markup language that allows you to define the tags(markup) needed to identify the data and text in XML documents.

Worker A worker is an out-of-process unthreaded procedure. The conversation to the backend system is done by the worker process. The worker returns the results to the connector using the proprietary protocol.

Index

P

R

S

T

U

V

W