BEA eLink Adapter for Portal Infranet

User Guide

BEA eLink Adapter for Portal Infranet 1.1
Document Edition 1.0
April 2000

Copyright

Restricted Rights Legend

Trademarks or Service Marks

BEA eLink Adapter for Portal Infranet User Guide

Document Edition	Part Number	Date	Software Version
1.0	N/A	January 2000	BEA eLink Adapter for Portal Infranet 1.0
1.1	N/A	April 2000	BEA eLink Adapter for Portal Infranet 1.1

Contents

4. Invoking Portal Infranet Services

5. Running the Sample Application

A. Portal Infranet Service Interfaces

About This Document

BEA eLink Adapter for Portal Infranet is an eLink Platform server that allows functionality within Portal Infranet to be invoked from the eLink Platform environment. The eLink Adapter for Portal Infranet advertises multiple Portal Infranet business functions in the form of eLink Platform services. The Portal Infranet functionality and the names under which they are advertised are configurable in the eLink Adapter for Portal Infranet's configuration file.

The *BEA eLink Adapter for Portal Infranet User Guide* is organized as follows:

- *BEA eLink Adapter for Portal Infranet Overview* introduces the eLink Adapter for Portal Infranet component and explains how eLink Adapter for Portal Infranet fits into the eLink Platform environment.
- *Installing the eLink Adapter for Portal Infranet* explains how to install the eLink Adapter for Portal Infranet component.
- *Configuring the eLink Adapter for Portal Infranet* provides information for configuring the eLink Adapter for Portal Infranet.
- *Invoking Portal Infranet Services* provides information for accessing the eLink to Portal Infranet services.
- *Running the Sample Application* provides information about how to build, configure, and execute the sample applications.
- *Portal Infranet Service Interfaces* provides a complete list of all of the interfaces you may use as you integrate the Portal Infranet services.
- *Portal Infranet Enumerations* provides a complete list of all of the enumerations you may use as you integrate the Portal Infranet services.
- *Error and Informational Messages* describes error and informational messages as well as actions to resolve the errors.

What You Need to Know

This document is intended for system administrators who will install the eLink Adapter for Portal Infranet on various platforms, as well as programmers who will configure the eLink Adapter for Portal Infranet and set up eLink Platform services to execute information transfers with Portal Infranet. This guide assumes knowledge of eLink Platform and Portal Infranet products.

e-docs Web Site

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

You can print a copy of this document from a Web browser, one file at a time, by using the File—>Print option on your Web browser. A PDF version of this document is available on the eLink Adapter for Portal Infranet documentation Home page on the e-docs Web site (and also on the documentation CD). You can open the PDF in Adobe Acrobat Reader and print the entire document (or a portion of it) in book format.

To access the PDFs, open the eLink Adapter for Portal Infranet documentation Home page, click the PDF files button and select the document you want to print. If you do not have the Adobe Acrobat Reader, you can get it for free from the Adobe Web site at http://www.adobe.com/.

Related Information

The following BEA publications are also available:

- *TUXEDO System 6 Reference Manual*
- *TUXEDO System 6 Programmer's Guide, Volumes 1 and 2*
- *TUXEDO System 6 FML Programmer's Guide*

Contact Us!

Your feedback on the eLink Adapter for Portal Infranet documentation is important to us. Send us e-mail at **docsupport@beasys.com** if you have questions or comments. Your comments will be reviewed directly by the BEA professionals who create and update the eLink Adapter for Portal Infranet documentation.

In your e-mail message, please indicate that you are using the documentation for the BEA eLink Adapter for Portal Infranet 5.0 release.

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

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

- Your name, e-mail address, phone number, and fax number
- Your company name and company address
- Your machine type and authorization codes
- The name and version of the eLink Adapter for Portal Infranet you are using
- A description of the problem and the content of pertinent error messages

Documentation Conventions

Convention	Item
boldface text	Indicates terms defined in the glossary.
Ctrl+Tab	Indicates that you must press two or more keys simultaneously.
italics	Indicates emphasis or book titles.

Convention	Item

`monospace text`	Indicates code samples, commands and their options, data structures and their members, data types, directories, and file names and their extensions. Monospace text also indicates text that you must enter from the keyboard.
	Examples:
	`#include <iostream.h> void main ( ) the pointer psz`
	`chmod u+w *`
	`\tux\data\ap`
	`.doc`
	`tux.doc`
	`BITMAP`
	`float`

`monospace boldface text`	Identifies significant words in code. *Example*: `void` **`commit`** `( )`
`monospace italic text`	Identifies variables in code. *Example*: `String` *`expr`*
UPPERCASE TEXT	Indicates device names, environment variables, and logical operators. *Example*s: LPT1 SIGNON OR
`{ }`	Indicates a set of choices in a syntax line. The braces themselves should never be typed.

`[ ]`	Indicates optional items in a syntax line. The brackets themselves should never be typed.
	Example:
	`buildobjclient [-v] [-o name ] [-f` *`file-list`*`]...`
	`[-l` *`file-list`*`]...`

`|`	Separates mutually exclusive choices in a syntax line. The symbol itself should never be typed.

Convention	Item

. . .	Indicates one of the following in a command line:
	■ That an argument can be repeated several times in a command line
	■ That the statement omits additional optional arguments
	■ That you can enter additional parameters, values, or other information
	The ellipsis itself should never be typed.
	Example:
	`buildobjclient [-v] [-o name ] [-f file-list]...`
	`[-l file-list]...`

. . .	Indicates the omission of items from a code example or from a syntax line. The vertical ellipsis itself should never be typed.

1 BEA eLink Adapter for Portal Infranet Overview

This chapter contains the following topics:

- BEA eLink Solution Overview
- BEA eLink Adapter Feature Overview
  - What is Portal Infranet?
  - What is eLink Adapter for Portal Infranet?

BEA eLink Solution Overview

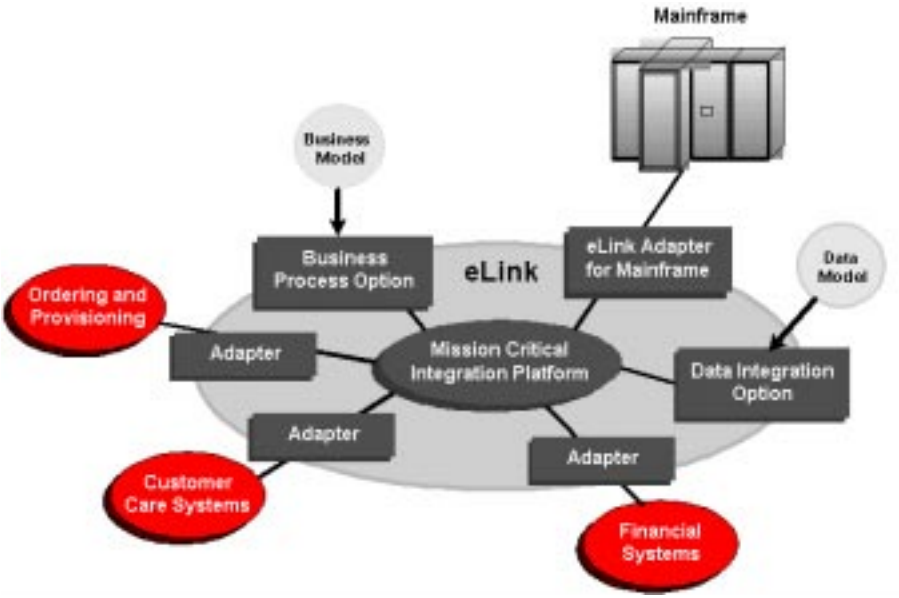
BEA eLink™ provides an open Enterprise Application Integration (EAI) solution that allows applications throughout organizations to communicate seamlessly. Using EAI, you gain the long-term flexibility and investment protection you need to keep up with today's ever-changing business environment.

Typically, companies use packaged applications to automate internal operations, such as financial, manufacturing, human resources, etc. While they successfully address the needs of these specific areas, these proprietary platforms often do not work together. To compete today, you need a much greater exchange of information. Systems need to communicate at a process level within your own organization, as well as with

customer's and supplier's systems. BEA eLink Platform is the underlying basis of BEA eLink™, a family of off-the-shelf enterprise application integration (EAI) products that leverage BEA's transaction platform to integrate existing legacy applications with customer-focused and business-to-business e-commerce initiatives.

BEA eLink Platform provides a proven infrastructure for integrating applications within the enterprise and across the Web. BEA eLink Platform ensures high-performance, secure transactions and transparent access to mission-critical applications and information throughout the enterprise and across the Web. Figure 1-1 illustrates the eLink logical architecture and shows where the eLink Adapters fit into the process.

Figure 1-1 BEA eLink Solution Illustration



The entire BEA eLink family (including all options and adapters) is highly scalable. Multiple instances of BEA eLink components can collaborate so that work is divided between eLink domains. BEA eLink includes SNMP integration for enterprise management.

The current BEA eLink Platform leverages the BEA TUXEDO infrastructure because it is based on a service-oriented architecture. Both BEA TUXEDO and BEA eLink communicate directly with each other and with other applications through the use of **services**. Multiple services are grouped into "application servers" or "servers". The

terms, TUXEDO services/ servers and eLink services/servers can be used interchangeably. Because this document is specifically addressing the eLink family, the term “eLink service” and “eLink server” is used throughout.

The BEA eLink Platform complies with the Open Group’s X/Open standards including support of the XA standard for two-phase commit processing, the X/Open **ATMI** API, and XPG standards for language internationalization. C, C++, COBOL, and Java are supported. The BEA eLink Platform connects to any RDBMS, OODBMS, file manager or queue manager, including a supplied XA-compliant queueing subsystem.

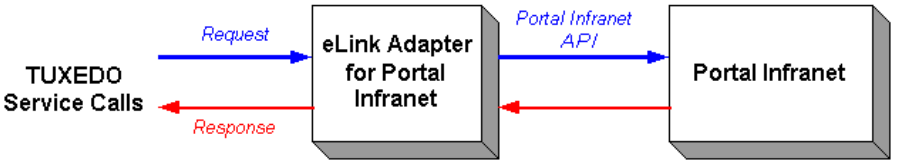
The following components operate with BEA eLink Platform:

- The **Data Integration Option** translates data models used by different applications into a common data format. It provides a cost-effective alternative to writing or generating programs to perform this function. It also handles complex translation with great power and scalability. The DIO leverages technology based on the TSI Mercator product, which is integrated with eLink.
- The **Business Process Option** helps automate tasks in the distributed global business process and dynamically responds to business events and exceptions. The BPO is currently implemented by integrating eLink with technology based on InConcert workflow management software.
- An **eLink Adapter** provides the interface between the BEA eLink Platform and external applications with out-of-the-box functionality.

BEA eLink Adapter Feature Overview

The eLink Adapter for Portal Infranet is an eLink Platform server that allows Portal Infranet functionality to be invoked in the form of services from a eLink Platform environment. Portal Infranet business services are advertised as eLink Platform services. Figure 1-2 shows the relationship of the eLink Adapter for Portal Infranet with eLink Platform and Portal Infranet.

Figure 1-2 The eLink to Portal Infranet Transfer Process



This section contains overview information for each of these topics:

- What is Portal Infranet?
- What is eLink Adapter for Portal Infranet?

What is Portal Infranet?

Portal Infranet is a customer management and billing system designed for IP-based businesses. It uses a database to integrate and store service information such as pricing models, business policies, and customer information. Portal Infranet supports the following business tasks:

- **Customer Registration**
  Information is collected from a customer, validated against the set business rules, and an account is created.
- **Customer Management Process**
  Modified information is collected, revaluated and stored in the Portal Infranet database and logged in the account for further reference *(audit trail).*

- **Authentication and Authorization Process**
  Portal Infranet authenticates users based on log-in and password, checks account status and authorizes access to individual services.
- **Activity Tracking Process**
  Portal Infranet tracks all of the activities (like login and logout) of the customer on a real-time basis.
- **Real-Time Rating Process**
  The rating process rates all of the activities from the activity tracking process and applies the rates based on the set pricing plan and updates the account.
- **Billing Process**
  This process collects all of the account information, charges the customer, and collects payment. This could be either by sending an invoice, charging a credit card, or by any other payment method.

What is eLink Adapter for Portal Infranet?

The eLink Adapter for Portal Infranet is an eLink Platform server that allows functionality within Portal Infranet to be invoked from the eLink Platform environment. Each unique Portal Infranet function that can be invoked is advertised as a unique eLink Platform service. This service can be invoked using request and response are FML32 buffers.

The eLink Adapter for Portal Infranet advertises multiple Portal Infranet business functions as eLink Platform services. These services are configurable in the adapter's configuration file. The services that the eLink Adapter for Portal Infranet offers may be called using the ATMI functions tpcall() or tpacall(). Both of these functions are request/response, but tpacall() is asynchronous while tpcall() is a synchronous invocation. The eLink Adapter for Portal Infranet advertises eLink Platform services for the following Portal Infranet business functions:

- **Creating an Account**
  Adding a customer account to the Portal Infranet billing system
- **Deleting a Product from an Account**
  Deleting a a product from an account
- **Modifying an Account**
  Modifying a customer's account information

- **Modifying Payment Info**
  Modifying a customer's payment information
- **Modifying Billing Info**
  Modifying the billing address of the customer
- **Modifying Credit Balance Info**
  Modifying a customer's credit balance
- **Modifying Locale Info**
  Modifying a customer's locale
- **Modifying Contact Info**
  Modifying contact information for a customer
- **Adding a New Contact**
  Adding contact information for a customer
- **Querying Account Information**
  Querying customer accounts
- **Querying Product Information**
  Querying product information
- **Querying Rate Information**
  Querying billing rate information

Note: For a complete list of Portal Infranet Service interfaces, please see Appendix A, "Portal Infranet Service Interfaces" of this User Guide.

2 Installing the eLink Adapter for Portal Infranet

This chapter contains the following topics:

- Installation Prerequisites
- Installing on the Windows NT 4.0 Platform
- Installing on the UNIX Platform
- Distribution Libraries and Executables

Installation Prerequisites

Refer to the *BEA eLink Adapter for Portal Infranet Release Notes* for information on prerequisite software that must be installed and operational prior to installing the eLink Adapter for Portal Infranet software.

Note: BEA eLink Platform 1.2 must be installed prior to installing the eLink Adapter for Portal Infranet component for the eLink Platform execution environment.

The current BEA eLink Platform leverages the BEA Tuxedo infrastructure because it is based on a service-oriented architecture. Both BEA Tuxedo and BEA eLink communicate directly with each other and with other applications through the use of

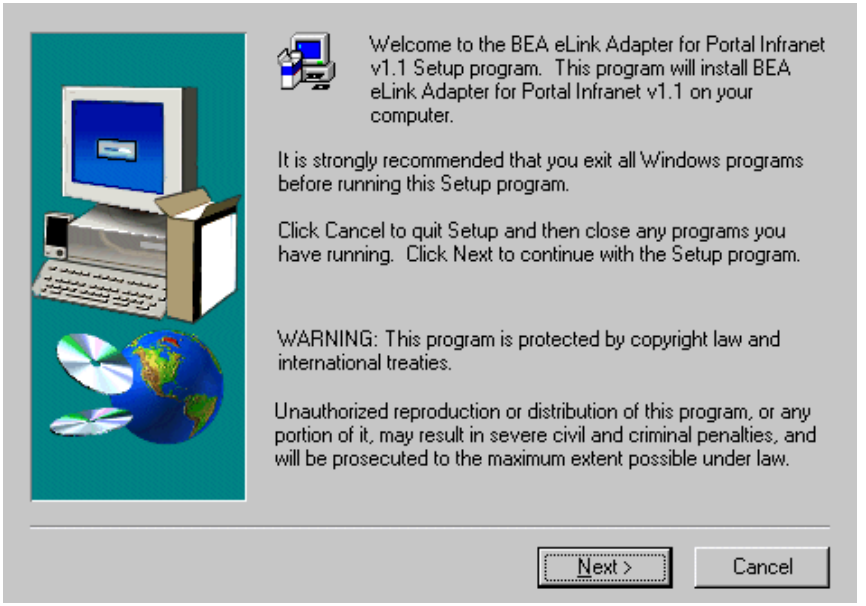
services. Multiple services are grouped into “application servers” or “servers”. The terms, Tuxedo services/ servers and eLink services/servers can be used interchangeably. Because this document is specifically addressing the eLink family, the term “eLink service” and “eLink server” is used throughout.

Installing on the Windows NT 4.0 Platform

Perform the following steps to install the eLink Adapter for Portal Infranet software on a Windows NT system:

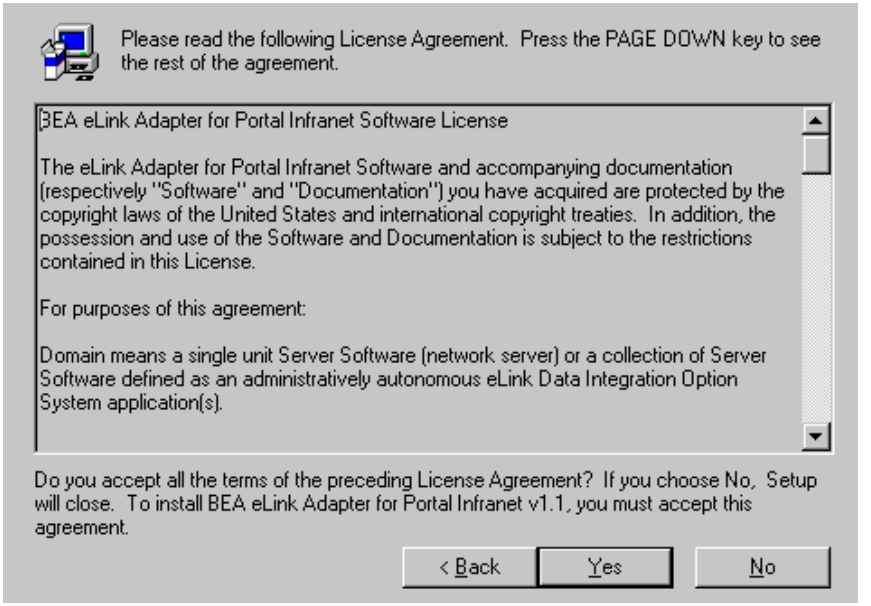
1. Insert the product CD-ROM and click on the **Run** option from the **Start menu**. The **Run** window displays. Click on the **Browse** button to select the CD-ROM drive. Select the `winnt` directory and select the `Setup.exe` program. Click **OK** to run the executable and begin the installation. The following **Welcome** window displays. Click **Next** to continue with the installation.

Figure 2-1 Welcome



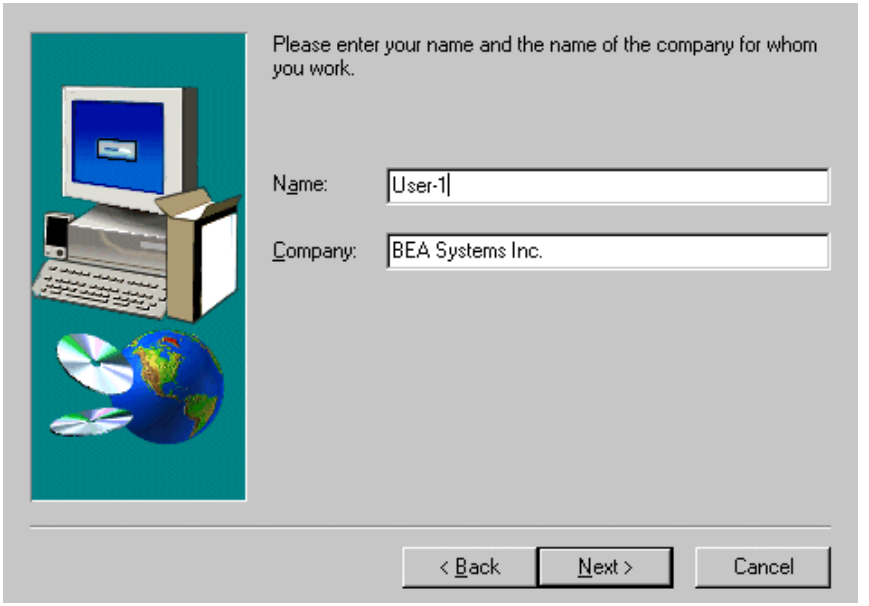
2. The **License Agreement** window displays after the **Welcome** window. Read the license agreement information, and click **Yes** to continue with the installation.

Figure 2-2 License Agreement



3. The **User Information** window displays after the **Welcome** window. Enter your name in the **Name** field. Enter the name of your company in the **Company** field. Click **Next** to continue with the installation.

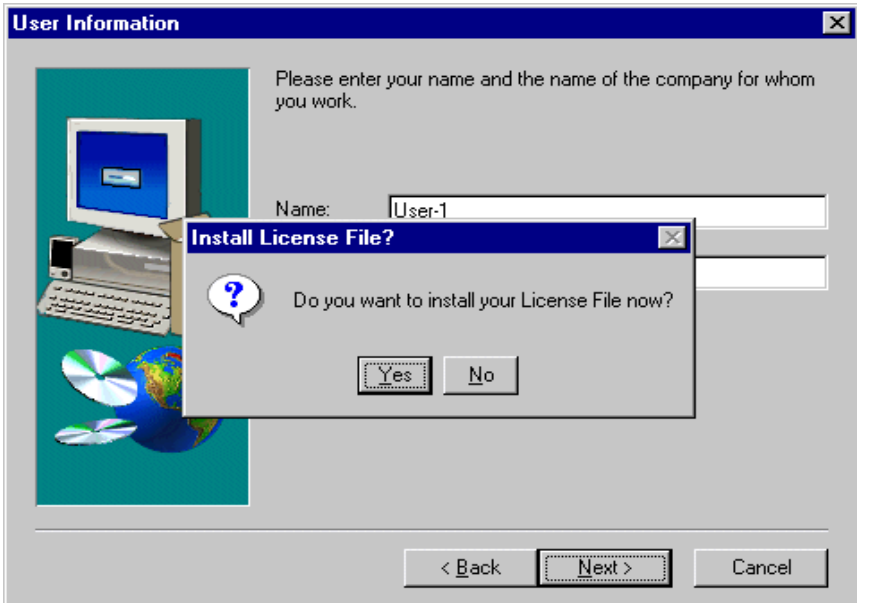
Figure 2-3 User Information



4. The **Install License File** pop-up window displays over the **User Information** window after you click **Next** as shown in Figure 2-4. Click the **Yes** or **No** button to continue the installation with or without the license file.

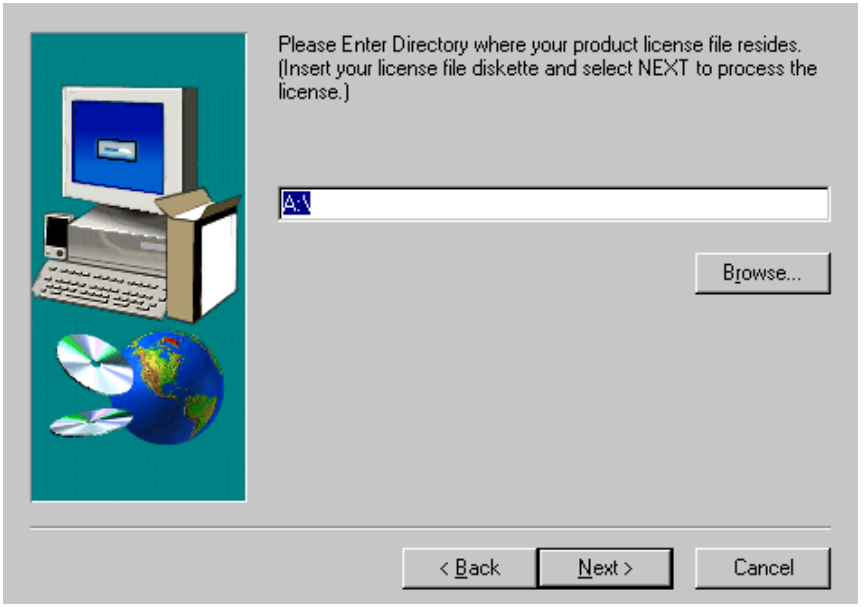
Note: For additional license key information, refer to the eLink Adapter for Portal Infranet *Release Notes*.

Figure 2-4 Install License Key Pop-Up Window



a. **Installing with the License File**
Click **Yes** in the Install License File pop-up window as shown in Figure 2-4 to install the license file. Click **Browse** to locate the License file as shown in Figure 2-5. Click **Next** to continue with the installation process.

Figure 2-5 License File Browser Window



A progress bar displays the status of the installation. You may abort the installation process anytime prior to completion by clicking the **Cancel** button.

b. **Installing without the License File**
Click **No** in the Install License File pop-up window as shown in Figure 2-4 to bypass the installation of the license file now. Be sure to install the license file prior to initializing the software.

A progress bar displays the status of the installation. You may abort the installation process anytime prior to completion by clicking the **Cancel** button.

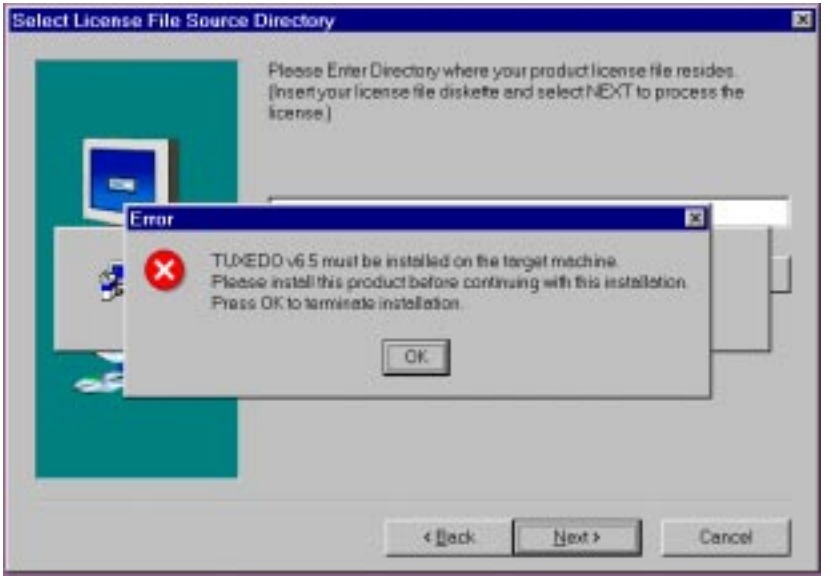
Note: If you select No, the installation continues but an error is generated in the ulog.mm/dd/yy file indicating that the product is unlicensed. Please refer to the "Using the License Key" section of the BEA eLink Adapter for Portal Infranet Release Notes for instructions on using the license file.

5. **If eLink Platform is already installed on your system:**
   If BEA eLink Platform is installed and detected on your system, the installation begins and a progress bar displays the status. The eLink Adapter components install into the eLink Platform directory. You may abort the installation process anytime prior to completion by clicking the **Cancel** button.

   When the installation completes, the **Setup Complete** window shown in Figure 2-7 notifies you that the eLink Adapter software is installed on your system.

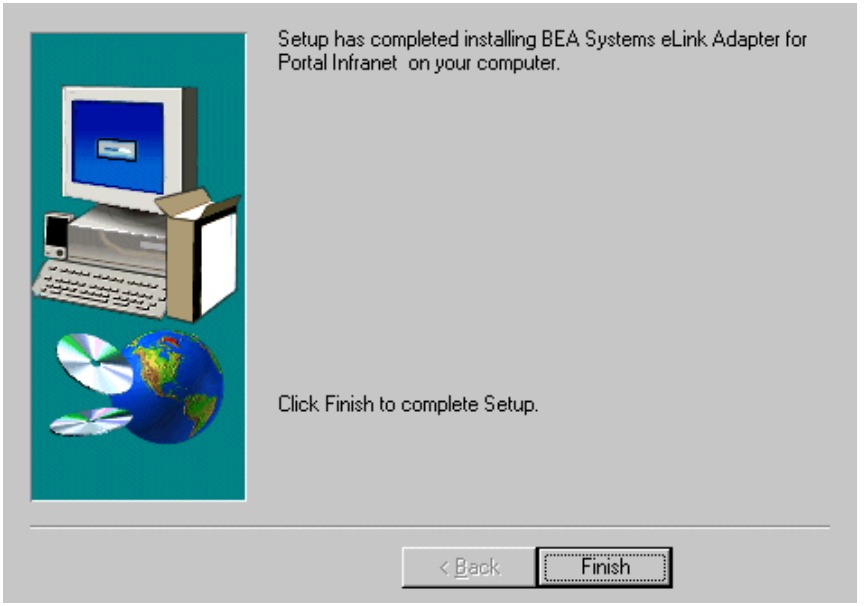
   If eLink Platform is NOT already installed on your system:
   If BEA eLink Platform is not installed on your system, the following **Error** pop-up window displays as shown in Figure 2-6. Click **OK** on the pop-up window to terminate the installation process. Install eLink Platform on your system (see warning above). Re-initiate the installation process starting with step one of these installation instructions.

Figure 2-6 eLink Platform Installation Error Pop-Up Window



6. The **Setup Complete** window notifies you that the eLink Adapter software is installed on your system. Click **Finish** to complete the setup process.

Figure 2-7 Setup Complete



Installing on the UNIX Platform

This section explains how to install the eLink Adapter software on the HP-UX 11.0, HP-UX 10-20, Solaris 2.6 and Solaris 7 execution platforms.

Warning: You must install the eLink Adapter for Portal Infranet execution components within the eLink Platform directory.

To install the eLink Adapter for Portal Infranet software, you run the `install.sh` script. This script installs all the necessary software components.

Note: The eLink Adapter for Portal Infranet should be installed into the eLink Platform directory. Prior to initiating the installation script, determine the directory location of eLink Platform.

Perform the following steps to install the eLink Adapter for Portal Infranet software on a supported Unix platform:

1. Log on as root to install the eLink Adapter for Portal Infranet software.

```
$ su -
Password:
```

2. Access the CD-ROM device.

```
# ls -1 /dev/cdrom

total 0

brw-rw-rw-     1 root     sys     22,   0 May 17 10:55 c1b0t010
```

3. Mount the CD-ROM.

```
# mount -r -F cdfs /dev/cdrom/c1b0t0l0 /mnt
```

4. Change the directory to your CD-ROM device.

```
# cd /mnt
```

5. List the CD-ROM contents.

```
# ls

install.sh  hp
```

6. Execute the installation script.

```
# sh ./install.sh
```

7. The installation script runs and prompts you for responses.

Listing 2-1 Install.sh Example

```
cmadm@dalhpw1:/cmhome/dist/blade-1> ls
hp/          install.sh*  sun5x/       winnt/
cmadm@dalhpw1:/cmhome/dist/blade-1> sh install.sh

01) hp/hpux1020      02) hp/hpux11        03) sun5x/sol26
04) sun5x/sol7
```

```
Install which platform's files? [01-4, q to quit, l for list]: 1

** You have chosen to install from hp/hpux1020 **

BEA eLink Adapter for Portal Release 1.1

This directory contains the BEA eLink Adapter for Portal System for

HP-UX 10.20 on 9000/800 series.


Is this correct? [y,n,q]: y

To terminate the installation at any time
press the interrupt key,
typically <del>, <break>, or <ctrl+c>.

The following packages are available:

  1     port            BEA eLink Adapter for Portal

Select the package(s) you wish to install (or 'all' to install
all packages) (default: all) [?,??,q]:

BEA eLink Adapter for Portal
(9000) Release 1.1
Copyright (c) 2000 BEA Systems, Inc.
All Rights Reserved.
Distributed under license by BEA Systems, Inc.
BEA eLink is a trademark of BEA Systems, Inc.


Directory where Portal Adapter files are to be installed
(Enter your eLink Platform directory path) [?,q]: /work/cmadm/tux65


Using /work/cmadm/tux65 as the Portal Adapter base directory

Determining if sufficient space is available ...
670 blocks are required
4332666 blocks are available to /work/cmadm/tux65

Unloading /cmhome/dist/blade-1/hp/hpux1020/port/PORTT65.Z ...
bin/ELINKPORTALO
bin/lic.sh
eLink/catalogs/ELINKPORTALO.txt
eLink/portal/simpportal/create_cust.c
eLink/portal/simpportal/elinkportal.cfg
```

```
eLink/portal/simpportal/elinkportal.env
eLink/portal/simpportal/elinkportal.ubb
eLink/portal/simpportal/pin.conf
eLink/portal/simpportal/prtconst.h
eLink/portal/simpportal/readme.txt
eLink/portal/simpportal/setenv.sh
eLink/portal/simpportal/udscripts//get_acc_info.ud
eLink/portal/simpportal/udscripts/add_contact.ud
eLink/portal/simpportal/udscripts/add_deal.ud
eLink/portal/simpportal/udscripts/add_service.ud
eLink/portal/simpportal/udscripts/create_cust.ud
eLink/portal/simpportal/udscripts/create_cust_cc.ud
eLink/portal/simpportal/udscripts/del_acc.ud
eLink/portal/simpportal/udscripts/del_product.ud
eLink/portal/simpportal/udscripts/get_bill_list.ud
eLink/portal/simpportal/udscripts/get_invo_img.ud
eLink/portal/simpportal/udscripts/get_line_items.ud
eLink/portal/simpportal/udscripts/get_prod_info.ud
eLink/portal/simpportal/udscripts/get_rate_info.ud
eLink/portal/simpportal/udscripts/invalid.ud
eLink/portal/simpportal/udscripts/login.ud
eLink/portal/simpportal/udscripts/logout.ud
eLink/portal/simpportal/udscripts/lookup.ud
eLink/portal/simpportal/udscripts/mod_acc_info.ud
eLink/portal/simpportal/udscripts/mod_acc_stat_active.ud
eLink/portal/simpportal/udscripts/mod_acc_stat_inactive.ud
eLink/portal/simpportal/udscripts/mod_bill_info.ud
eLink/portal/simpportal/udscripts/mod_con_info.ud
eLink/portal/simpportal/udscripts/mod_cred_bal1.ud
eLink/portal/simpportal/udscripts/mod_cred_bal2.ud
eLink/portal/simpportal/udscripts/mod_cred_bal3.ud
eLink/portal/simpportal/udscripts/mod_loc_info.ud
eLink/portal/simpportal/udscripts/mod_pay_info.ud
eLink/portal/simpportal/udscripts/mod_prod_stat.ud
eLink/portal/simpportal/udscripts/mod_product.ud
eLink/portal/simpportal/udscripts/mod_serv_active.ud
eLink/portal/simpportal/udscripts/mod_serv_inactive.ud
eLink/portal/simpportal/udscripts/mod_serv_log.ud
eLink/portal/simpportal/udscripts/mod_serv_pass.ud
eLink/portal/simpportal/udscripts/post_activity.ud
eLink/portal/simpportal/udscripts/validate.ud
eLink/portal/simpportal/udscripts/who.ud
lib/libadk.sl.1.02
udataobj/elinkportal.fml
600 blocks
... finished

Changing file permissions...
... finished
```

```
If your license file is accessible, you may install it now.
Install license file? [y/n]: n

Please don't forget to use lic.sh located in your product bin
directory
to install the license file from the enclosed floppy.
Refer to your product Release Notes for details on how to do this.


Installation of BEA eLink Adapter for Portal was successful

Please don't forget to fill out and send in your registration card

cmadm@dalhpw1:/cmhome/dist/blade-1>
```

8. Change the directory to your root directory.

```
# cd /
```

9. Unmount the CD-ROM device.

```
# unmount /mnt
```

Distribution Libraries and Executables

The eLink Adapter for Portal Infranet CD-ROM contains the following libraries and executable programs. After installing the eLink Adapter for Portal Infranet software, verify that these libraries and programs are installed on your system. Verify that the following files are installed by the eLink Adapter for Portal Infranet software. These files should be installed in the listed subdirectories relative to `$(TUXDIR)`.

Table 2-1 eLink Adapter for Portal Infranet 1.1 Platform Support

Operating System	Binary Port
HP-UX 11.00	Native
HP-UX 10.20	Native
Solaris 2.6	Native
Solaris 7	Native Solaris 2.6
NT 4.0	Native

Table 2-2 Installation Files and Directories

bin	eLink\portal\ simpportal	eLink\portal\ simpportal\udscripts	eLink\ catalogs	lib	udataobj
ELINKPORTALO	readme.txt	add_contact.ud	ELINKPORTALO.TXT	libadk.sl(HP)	elinkportal.fml
	elinkportal.cfg	add_deal.ud		libadk.dll(NT)	
	elinkportal.env	add_service.ud			
	elinkportal.ubb	create_cust_ud			
	pin.conf	create_cust_cc.ud			
	setenv.sh	del_acc.ud			
	create-cust.c	del_product.ud			

bin	eLink\portal\ simpportal	eLink\portal\ simpportal\udscripts	eLink\ catalogs	lib	udataobj
	pin.cnf (NT)	get_acc_info.ud			
	prtconst.h	get_bill_list.ud			
	cetenv.bat(NT)	get_invo_img.ud			
		get_line_items.ud			
		get_prod_info.ud			
		get_rate_info.ud			
		invalid.ud			
		login.ud			
		logout.ud			
		lookup.ud			
		mod_acc-info.ud			
		mod-acc_stat_ active.ud			
		mod-acc_stat_ inactive.ud			
		mod_bill_info.ud			
		mod_con_info.ud			
		mod_cred_bal1.ud			
		mod_cred_bal2.ud			
		mod_cred_bal3.ud			
		mod_loc_info.ud			
		mod_pay_info.ud			
		mod_prod_stat.ud			
		mod_product.ud			
		mod_serv_active. ud			
		mod_serv_inactive. ud			
		mod_serv_log.ud			
		mod_serv_pass.ud			
		post_activity.ud			
		validate.ud			
		who.ud			

3 Configuring the eLink Adapter for Portal Infranet

This chapter consists of the following topics:

- Adding the ELINKPORTALO Server to the UBBCONFIG File
- Modifying the Adapter Configuration File
- Modifying the Portal Infranet Configuration File

Adding the ELINKPORTALO Server to the UBBCONFIG File

The `ELINKPORTALO` server must be added to the `UBBCONFIG` file so that it is part of your eLink Platform system. To define this server, add `ELINKPORTALO` to the SERVERS section of the `UBBCONFIG` file. You must specify the `CLOPT` parameter to indicate the name of the configuration file that the adapter uses at run-time. The `CLOPT` parameter should be specified as:

```
CLOPT="-- -C elinkportal.cfg"
```

Listing 3-1 shows the syntax for the server definition in the `UBBCONFIG` file.

Listing 3-1 Syntax for ELINKPORTALO Server Definition in the UBBCONFIG File

```
*SERVERS

ELINKPORTALO
       SRVGRP=groupname     SRVID=n
       CLOPT="-- -C elinkportal.cfg"
```

For more information about the SRVGRP, SRVID, and CLOPT parameter syntax and definitions, refer to the *BEA TUXEDO Reference Manual.*

Listing 3-2 is a sample UBBCONFIG file. In this sample, the ELINKPORTALO server is defined in the SERVERS section with the required CLOPT option specified.

Listing 3-2 Sample UBBCONFIG File for Portal Infranet

```
*RESOURCES

IPCKEY        123791
DOMAINID      simpapp
MASTER        simple

*MACHINES

DAL1
       LMID= simple
       TUXDIR= "/tuxedo"
       TUXCONFIG= "/myappdir/tuxconfig"
       APPDIR= "/myappdir"
       FIELDTBL = "sample.fml"
       FIELDTBL32= "sample.fml"
       FLDTBLDIR= "/myappdir"
       FLDTBLDIR32= "/myappdir"
       ULOGPFX= "/myappdir/ULOG"


*GROUPS

eLINK
         LMID=simple GRPNO=1

*SERVERS
```

```
DEFAULT:
       CLOPT="-A"

ELINKPORTALO
              SRVGRP=eLINK  SRVID=10
              CLOPT="-- -C elinkportal.cfg"


*SERVICES

*ROUTING
```

Modifying the Adapter Configuration File

The elink Adapter for Portal Infranet reads a configuration file at startup to advertise services that retrieve or update data maintained by the Portal Infranet System. This configuration file contains ASCII text that describes the services that are to be advertised and what these services manipulate within Portal Infranet. The name of the configuration file is arbitrarily chosen by the user, but it must match what is specified in the `UBBCONFIG` file (as described in the Adding the ELINKPORTALO Server to the UBBCONFIG File section of this guide). This configuration file must be located in the application directory (`APPDIR`) of the end-user's application.

The adapter configuration file contains three different types of sections. Each section defines specific configuration parameters. The SERVER section defines the global flags that manipulate the operation of the adapter. SERVICE sections define each service that is advertised by the adapter. FIELD_GROUP sections define groups of fields that are input or output of a specified service. Each section is begins with by a text label prefixed with an asterisk (e.g. *SERVICE).

The only part of the configuration file that should be modified is the SERVER section. The definitions in the provided sample configuration file define services for account management and billing via Portal Infranet. These definitions do not need to be modified unless you wish to change the name of the advertised service. Nevertheless, all configuration sections and parameters are described below.

The SERVER Section

In this section you can specify NAME, EXIT_CONNECT_LOSS, MINMSGLEVEL, and MAXMSGLEVEL parameters. The following table provides descriptions for these parameters.

Service Name	Description
`NAME` = < Name >	A unique identifier for a specific instance of the eLink Adapter for Portal Infranet. This value should be alphanumeric text.

Service Name	Description
EXIT_CONNECT_LOSS = < Y/N >	Specifies whether the eLink Adapter for Portal Infranet should exit if the connection with Portal Infranet is lost.
MINMSGLEVEL and MAXMSGLEVEL	Specifies that the adapter should log additional diagnostic information in order to troubleshoot problems. The two parameters are used to specify a range of trace messages to be logged. The range 1 to 9 specifies that all diagnostic messages should be logged.

The SERVICE Section

For each service that is to be advertised by the adapter, a SERVICE section defines service functionality within Portal Infranet. Each SERVICE section defines the NAME of the service to be advertised, the OPERATION to be performed in the Portal Infranet system, the PORTAL_OPCODE to be executed, and the INPUT_MESSAGE and OUTPUT_MESSAGE of the service.

Service Name	Description
NAME = < Name >	The name of the eLink Platform service to be advertised by the adapter.
OPERATION = < CREATE / MODIFY / DELETE / GET >	The operation that is to be performed within the Portal Infranet database. For example a service that creates a customer would be defined with an OPERATION of CREATE.
PORTAL_OPCODE = PCM_OP_xxx	The Portal Infranet PCM_OP() code that is to be executed by this service. See Portal's Programmer's Guide for a list of valid Portal Infranet PCM_OP() codes.
INPUT_MESSAGE = < Field Group Name >	The input message expected by this service is further defined in a FIELD_GROUP section. The name of this FIELD_GROUP is specified in the INPUT_MESSAGE parameter.
OUTPUT_MESSAGE = < Field Group Name >	The output message returned by this service is further defined in a FIELD_GROUP section. The name of this FIELD_GROUP is specified in the OUTPUT_MESSAGE parameter.

The FIELD_GROUP Section

A FIELD_GROUP defines the fields that make up a request or response message of a service. A FIELD_GROUP lists the fields and subgroups of fields that are to be sent or received from Portal Infranet. A FIELD_GROUP definition may reference another FIELD_GROUP definition in order to define a sub-group hierarchy.

Service Name	Description
NAME = < Name >	The unique name of the FIELD_GROUP. This name may be referenced in the INPUT_MESSAGE or OUTPUT_MESSAGE parameters of a SERVICE definition, or may be referenced in another FIELD_GROUP definition.
FLIST_NAME= < FLIST Name >	The name of the FLIST that is to be created when this FIELD_GROUP is translated to a Portal Infranet FLIST data structure.
OCCRANCE_FIELD= < Field Name >	The name of the field that explicitly denotes how many occurrences of this FIELD_GROUP exist in a given message. See Chapter 4 on Hierarchical Data.
FIELDS= < Field Name, Field Name, … >	List of names of the FML fields (without the FLD_ prefix) that make up this FIELD_GROUP. FIELD_GROUP names may also be referenced in the FIELDS list.

Listing 3-3 shows an example of the adapter configuration file that is provided with the Portal Infranet Adapter. The concepts discussed above are illustrated in this file.

Listing 3-3 The Adapter Configuration File

```
*SERVER
NAME=PORTAL_ADAPTER
EXIT_CONNECT_LOSS=Y
MINMSGLEVEL=1
MAXMSGLEVEL=9

*SERVICE
NAME=CREATE_CUST
OPERATION=CREATE
INPUT_MESSAGE=ACCOUNT_INPUT
```

```
OUTPUT_MESSAGE=ACCOUNT_OUTPUT

*SERVICE
NAME=DEL_PRODUCT
OPERATION=DELETE
INPUT_MESSAGE=DELPROD_INPUT
OUTPUT_MESSAGE=DELPROD_OUTPUT

*SERVICE
NAME=MOD_PAY_INFO
OPERATION=MODIFY
INPUT_MESSAGE=MODPAYINFO_INPUT
OUTPUT_MESSAGE=MODPAYINFO_OUTPUT

*SERVICE
NAME=MOD_BILL_INFO
OPERATION=MODIFY
INPUT_MESSAGE=MODBILLINFO_INPUT
OUTPUT_MESSAGE=MODBILLINFO_OUTPUT

*SERVICE
NAME=MOD_CRED_BAL
OPERATION=MODIFY
INPUT_MESSAGE=MODCREDBAL_INPUT
OUTPUT_MESSAGE=MODCREDBAL_OUTPUT

*SERVICE
NAME=MOD_ACC_INFO
OPERATION=MODIFY
INPUT_MESSAGE=MODACCINFO_INPUT
OUTPUT_MESSAGE=MODACCINFO_OUTPUT

*SERVICE
NAME=MOD_LOC_INFO
OPERATION=MODIFY
INPUT_MESSAGE=MODLOCINFO_INPUT
OUTPUT_MESSAGE=MODLOCINFO_OUTPUT

*SERVICE
NAME=MOD_CON_INFO
OPERATION=MODIFY
INPUT_MESSAGE=MODCONINFO_INPUT
OUTPUT_MESSAGE=MODCONINFO_OUTPUT

*SERVICE
NAME=ADD_CONTACT
OPERATION=MODIFY
INPUT_MESSAGE=ADDCON_INPUT
OUTPUT_MESSAGE=ADDCON_OUTPUT
```

```
*SERVICE
NAME=MOD_ACC_STAT
OPERATION=MODIFY
INPUT_MESSAGE=MODACCSTAT_INPUT
OUTPUT_MESSAGE=MODACCSTAT_OUTPUT


*SERVICE
NAME=ADD_DEAL
OPERATION=MODIFY
INPUT_MESSAGE=ADDDEAL_INPUT
OUTPUT_MESSAGE=ADDDEAL_OUTPUT

*SERVICE
NAME=MOD_PRODUCT
OPERATION=MODIFY
INPUT_MESSAGE=MODPROD_INPUT
OUTPUT_MESSAGE=MODPROD_OUTPUT


*SERVICE
NAME=ADD_SERVICE
OPERATION=MODIFY
INPUT_MESSAGE=ADDSERV_INPUT
OUTPUT_MESSAGE=ADDSERV_OUTPUT


*SERVICE
NAME=MOD_PROD_STAT
OPERATION=MODIFY
INPUT_MESSAGE=MODPRODSTAT_INPUT
OUTPUT_MESSAGE=MODPRODSTAT_OUTPUT


*SERVICE
NAME=MOD_SERV_LOG
OPERATION=MODIFY
INPUT_MESSAGE=MODSERVLOG_INPUT
OUTPUT_MESSAGE=MODSERVLOG_OUTPUT

*SERVICE
NAME=MOD_SERV_PASS
OPERATION=MODIFY
INPUT_MESSAGE=MODSERVPASS_INPUT
OUTPUT_MESSAGE=MODSERVPASS_OUTPUT

*SERVICE
NAME=MOD_SERV_STAT
```

```
OPERATION=MODIFY
INPUT_MESSAGE=MODSERVSTAT_INPUT
OUTPUT_MESSAGE=MODSERVSTAT_OUTPUT

*SERVICE
NAME=GET_BILL_LIST
OPERATION=GET
INPUT_MESSAGE=GETBILLLIST_INPUT
OUTPUT_MESSAGE=GETBILLLIST_OUTPUT

*SERVICE
NAME=GET_LINE_ITEMS
OPERATION=GET
INPUT_MESSAGE=GETLINEITEMS_INPUT
OUTPUT_MESSAGE=GETLINEITEMS_OUTPUT

*SERVICE
NAME=GET_INVO_IMG
OPERATION=GET
INPUT_MESSAGE=GETINVOIMG_INPUT
OUTPUT_MESSAGE=GETINVOIMG_OUTPUT

*SERVICE
NAME=GET_ACC_INFO
OPERATION=GET
INPUT_MESSAGE=GETACCINFO_INPUT
OUTPUT_MESSAGE=GETACCINFO_OUTPUT

*SERVICE
NAME=GET_PROD_INFO
OPERATION=GET
INPUT_MESSAGE=GETPRODINFO_INPUT
OUTPUT_MESSAGE=GETPRODINFO_OUTPUT

*SERVICE
NAME=GET_RATE_INFO
OPERATION=GET
INPUT_MESSAGE=GETRATEINFO_INPUT
OUTPUT_MESSAGE=GETRATEINFO_OUTPUT

*SERVICE
NAME=GET_AUTH
OPERATION=GET
INPUT_MESSAGE=GETAUTHINFO_INPUT
OUTPUT_MESSAGE=GETAUTHINFO_OUTPUT

*SERVICE
NAME=GET_ACCOUNT_NO
OPERATION=GET
```

```
INPUT_MESSAGE=GET_ACCOUNT_NO_INPUT
OUTPUT_MESSAGE=GET_ACCOUNT_NO_OUTPUT

*SERVICE
NAME=POST_ACTIVITY
OPERATION=MODIFY
INPUT_MESSAGE=POST_ACTIVITY_INPUT
OUTPUT_MESSAGE=POST_ACTIVITY_OUTPUT

*SERVICE
NAME=LOGIN
OPERATION=MODIFY
INPUT_MESSAGE=LOGIN_INPUT
OUTPUT_MESSAGE=LOGIN_OUTPUT

*SERVICE
NAME=LOGOUT
OPERATION=MODIFY
INPUT_MESSAGE=LOGOUT_INPUT
OUTPUT_MESSAGE=LOGOUT_OUTPUT

*SERVICE
NAME=GET_SESSIONS
OPERATION=MODIFY
INPUT_MESSAGE=GET_SESSIONS_INPUT
OUTPUT_MESSAGE=GET_SESSIONS_OUTPUT

*SERVICE
NAME=DEL_CUST
OPERATION=DELETE
INPUT_MESSAGE=DEL_CUST_INPUT
OUTPUT_MESSAGE=DEL_CUST_OUTPUT

*FIELD_GROUP
NAME=ACCOUNT_INPUT
FLIST_NAME=ACCOUNT_INPUT
OCCRANCE_FLD=ACCOUNT_INFO_OCC
FIELDS=NAME,SERVICES,BILLINFO,ACTGINFO,PAYINFO,NAMEINFO,EXEMPTION
S,LOCALES

*FIELD_GROUP
NAME=ACCOUNT_OUTPUT
FLIST_NAME=ACCOUNT_OUTPUT
OCCRANCE_FLD=ACCOUNT_INFO_OCC
FIELDS=OPS_ERROR,DESCR,ACCOUNT_NO,SERVICES,BILLINFO,ACTGINFO,PAYI
NFO,NAMEINFO,EXEMPTIONS,LOCALES

*FIELD_GROUP
NAME=DELPROD_INPUT
```

```
FLIST_NAME=DELPROD_INPUT
OCCRANCE_FLD=DELPROD_INFO_OCC
FIELDS=ACCOUNT_NO,PROD_NAME,ELEMENT_ID,QUANTITY

*FIELD_GROUP
NAME=DELPROD_OUTPUT
FLIST_NAME=DELPROD_OUTPUT
OCCRANCE_FLD=DELPROD_INFO_OCC
FIELDS=OPS_ERROR,DESCR

*FIELD_GROUP
NAME=MODPAYINFO_INPUT
FLIST_NAME=MODPAYINFO_INPUT
OCCRANCE_FLD=MODPAYINFO_INFO_OCC
FIELDS=ACCOUNT_NO,PAYINFO

*FIELD_GROUP
NAME=MODPAYINFO_OUTPUT
FLIST_NAME=MODPAYINFO_OUTPUT
OCCRANCE_FLD=MODPAYINFO_INFO_OCC
FIELDS=OPS_ERROR,DESCR

*FIELD_GROUP
NAME=MODBILLINFO_INPUT
FLIST_NAME=MODBILLINFO_INPUT
OCCRANCE_FLD=MODBILLINFO_INFO_OCC
FIELDS=ACCOUNT_NO,BILLINFO_2

*FIELD_GROUP
NAME=MODBILLINFO_OUTPUT
FLIST_NAME=MODBILLINFO_OUTPUT
OCCRANCE_FLD=MODBILLINFO_INFO_OCC
FIELDS=OPS_ERROR,DESCR

*FIELD_GROUP
NAME=MODCREDBAL_INPUT
FLIST_NAME=MODCREDBAL_INPUT
OCCRANCE_FLD=MODCREDBAL_INFO_OCC
FIELDS=ACCOUNT_NO,LIMIT

*FIELD_GROUP
NAME=MODCREDBAL_OUTPUT
FLIST_NAME=MODCREDBAL_OUTPUT
OCCRANCE_FLD=MODBILLINFO_INFO_OCC
FIELDS=OPS_ERROR,DESCR

*FIELD_GROUP
NAME=MODACCINFO_INPUT
FLIST_NAME=MODACCINFO_INPUT
```

```
OCCRANCE_FLD=MODACCINFO_INFO_OCC
FIELDS=ACCOUNT_NO,ACTGINFO

*FIELD_GROUP
NAME=MODACCINFO_OUTPUT
FLIST_NAME=MODACCINFO_OUTPUT
OCCRANCE_FLD=MODACCINFO_INFO_OCC
FIELDS=OPS_ERROR,DESCR


*FIELD_GROUP
NAME=MODLOCINFO_INPUT
FLIST_NAME=MODLOCINFO_INPUT
OCCRANCE_FLD=MODLOCINFO_INFO_OCC
FIELDS=ACCOUNT_NO,LOCALES

*FIELD_GROUP
NAME=MODLOCINFO_OUTPUT
FLIST_NAME=MODLOCINFO_OUTPUT
OCCRANCE_FLD=MODLOCINFO_INFO_OCC
FIELDS=OPS_ERROR,DESCR

*FIELD_GROUP
NAME=MODCONINFO_INPUT
FLIST_NAME=MODCONINFO_INPUT
OCCRANCE_FLD=MODCONINFO_INFO_OCC
FIELDS=ACCOUNT_NO,NAMEINFO

*FIELD_GROUP
NAME=MODCONINFO_OUTPUT
FLIST_NAME=MODCONINFO_OUTPUT
OCCRANCE_FLD=MODCONINFO_INFO_OCC
FIELDS=OPS_ERROR,DESCR

*FIELD_GROUP
NAME=ADDCON_INPUT
FLIST_NAME=ADDCON_INPUT
OCCRANCE_FLD=ADDCON_INFO_OCC
FIELDS=ACCOUNT_NO,NAMEINFO

*FIELD_GROUP
NAME=ADDCON_OUTPUT
FLIST_NAME=ADDCON_OUTPUT
OCCRANCE_FLD=ADDCON_INFO_OCC
FIELDS=OPS_ERROR,DESCR

*FIELD_GROUP
NAME=MODACCSTAT_INPUT
FLIST_NAME=MODACCSTAT_INPUT
```

```
OCCRANCE_FLD=MODACCSTAT_INFO_OCC
FIELDS=ACCOUNT_NO,DESCR,STATUS,STATUS_FLAGS,CLOSE_WHEN_T

*FIELD_GROUP
NAME=MODACCSTAT_OUTPUT
FLIST_NAME=MODACCSTAT_OUTPUT
OCCRANCE_FLD=MODACCSTAT_INFO_OCC
FIELDS=OPS_ERROR,DESCR

*FIELD_GROUP
NAME=ADDDEAL_INPUT
FLIST_NAME=ADDDEAL_INPUT
OCCRANCE_FLD=ADDDEAL_INFO_OCC
FIELDS=ACCOUNT_NO,DEAL_NAME

*FIELD_GROUP
NAME=ADDDEAL_OUTPUT
FLIST_NAME=ADDDEAL_OUTPUT
OCCRANCE_FLD=ADDDEAL_INFO_OCC
FIELDS=OPS_ERROR,DESCR


*FIELD_GROUP
NAME=MODPROD_INPUT
FLIST_NAME=MODPROD_INPUT
OCCRANCE_FLD=MODPROD_INFO_OCC
FIELDS=ACCOUNT_NO,PURCHASE_DISCOUNT,USAGE_DISCOUNT,PURCHASE_START
_T,PURCHASE_END_T,USAGE_START_T,USAGE_END_T,ELEMENT_ID,PROD_NAME

*FIELD_GROUP
NAME=MODPROD_OUTPUT
FLIST_NAME=MODPROD_OUTPUT
OCCRANCE_FLD=MODPROD_INFO_OCC
FIELDS=OPS_ERROR,DESCR

*FIELD_GROUP
NAME=ADDSERV_INPUT
FLIST_NAME=ADDSERV_INPUT
OCCRANCE_FLD=ADDSERV_INFO_OCC
FIELDS=ACCOUNT_NO,NAME,SERVICES

*FIELD_GROUP
NAME=ADDSERV_OUTPUT
FLIST_NAME=ADDSERV_OUTPUT
OCCRANCE_FLD=ADDSERV_INFO_OCC
FIELDS=OPS_ERROR,DESCR

*FIELD_GROUP
NAME=MODPRODSTAT_INPUT
```

```
FLIST_NAME=MODPRODSTAT_INPUT
OCCRANCE_FLD=MODPRODSTAT_INFO_OCC
FIELDS=ACCOUNT_NO,STATUS,PROD_NAME,ELEMENT_ID,STATUS_FLAGS

*FIELD_GROUP
NAME=MODPRODSTAT_OUTPUT
FLIST_NAME=MODPRODSTAT_OUTPUT
OCCRANCE_FLD=MODPRODSTAT_INFO_OCC
FIELDS=OPS_ERROR,DESCR

*FIELD_GROUP
NAME=MODSERVLOG_INPUT
FLIST_NAME=MODSERVLOG_INPUT
OCCRANCE_FLD=MODSERVLOG_INFO_OCC
FIELDS=ACCOUNT_NO,SERVICE_TYPE,LOGIN,NAME

*FIELD_GROUP
NAME=MODSERVLOG_OUTPUT
FLIST_NAME=MODSERVLOG_OUTPUT
OCCRANCE_FLD=MODSERVLOG_INFO_OCC
FIELDS=OPS_ERROR,DESCR

*FIELD_GROUP
NAME=MODSERVPASS_INPUT
FLIST_NAME=MODSERVPASS_INPUT
OCCRANCE_FLD=MODSERVPASS_INFO_OCC
FIELDS=ACCOUNT_NO,SERVICE_TYPE,LOGIN,PASSWD_CLEAR

*FIELD_GROUP
NAME=MODSERVPASS_OUTPUT
FLIST_NAME=MODSERVPASS_OUTPUT
OCCRANCE_FLD=MODSERVPASS_INFO_OCC
FIELDS=OPS_ERROR,DESCR

*FIELD_GROUP
NAME=MODSERVSTAT_INPUT
FLIST_NAME=MODSERVSTAT_INPUT
OCCRANCE_FLD=MODSERVSTAT_INFO_OCC
FIELDS=ACCOUNT_NO,SERVICE_TYPE,LOGIN,DESCR,STATUS,STATUS_FLAGS,CL
OSE_WHEN_T

*FIELD_GROUP
NAME=MODSERVSTAT_OUTPUT
FLIST_NAME=MODSERVSTAT_OUTPUT
OCCRANCE_FLD=MODSERVSTAT_INFO_OCC
FIELDS=OPS_ERROR,DESCR

*FIELD_GROUP
NAME=GETBILLLIST_INPUT
```

```
FLIST_NAME=GETBILLLIST_INPUT
OCCRANCE_FLD=GETBILLLIST_INFO_OCC
FIELDS=ACCOUNT_NO

*FIELD_GROUP
NAME=GETBILLLIST_OUTPUT
FLIST_NAME=GETBILLLIST_OUTPUT
OCCRANCE_FLD=GETBILLLIST_INFO_OCC
FIELDS=OPS_ERROR,DESCR,ACCOUNT_NO,BILLS_1

*FIELD_GROUP
NAME=GETLINEITEMS_INPUT
FLIST_NAME=GETLINEITEMS_INPUT
OCCRANCE_FLD=GETLINEITEMS_INFO_OCC
FIELDS=ACCOUNT_NO,BILLS

*FIELD_GROUP
NAME=GETLINEITEMS_OUTPUT
FLIST_NAME=GETLINEITEMS_OUTPUT
OCCRANCE_FLD=GETLINEITEMS_INFO_OCC
FIELDS=OPS_ERROR,DESCR,ACCOUNT_NO,BILL_NO,ITEMS

*FIELD_GROUP
NAME=GETINVOIMG_INPUT
FLIST_NAME=GETINVOIMG_INPUT
OCCRANCE_FLD=GETINVOIMG_INFO_OCC
FIELDS=ACCOUNT_NO,BILLS_2

*FIELD_GROUP
NAME=GETINVOIMG_OUTPUT
FLIST_NAME=GETINVOIMG_OUTPUT
OCCRANCE_FLD=GETINVOIMG_INFO_OCC
FIELDS=OPS_ERROR,DESCR,ACCOUNT_NO,BILLS_3

*FIELD_GROUP
NAME=GETACCINFO_INPUT
FLIST_NAME=GETACCINFO_INPUT
OCCRANCE_FLD=GETACCINFO_INFO_OCC
FIELDS=ACCOUNT_NO

*FIELD_GROUP
NAME=GETACCINFO_OUTPUT
FLIST_NAME=GETACCINFO_OUTPUT
OCCRANCE_FLD=GETACCINFO_INFO_OCC
FIELDS=OPS_ERROR,DESCR,ACCOUNT_NO,BALANCES,PRODUCTS,EXEMPTIONS,LO
CALES,SERVICES_1,BILLINFO_1,ACTGINFO_1,PAYINFO,NAMEINFO

*FIELD_GROUP
NAME=GETPRODINFO_INPUT
```

```
FLIST_NAME=GETPRODINFO_INPUT
OCCRANCE_FLD=GETPRODINFO_INFO_OCC
FIELDS=ACCOUNT_NO

*FIELD_GROUP
NAME=GETPRODINFO_OUTPUT
FLIST_NAME=GETPRODINFO_OUTPUT
OCCRANCE_FLD=GETPRODINFO_INFO_OCC
FIELDS=OPS_ERROR,DESCR,PRODUCTS_1

*FIELD_GROUP
NAME=GETRATEINFO_INPUT
FLIST_NAME=GETRATEINFO_INPUT
OCCRANCE_FLD=GETRATEINFO_INFO_OCC
FIELDS=ACCOUNT_NO,PROD_NAME,ELEMENT_ID

*FIELD_GROUP
NAME=GETRATEINFO_OUTPUT
FLIST_NAME=GETRATEINFO_OUTPUT
OCCRANCE_FLD=GETRATEINFO_INFO_OCC
FIELDS=OPS_ERROR,DESCR,RATES

*FIELD_GROUP
NAME=SERVICES
FLIST_NAME=SERVICES
OCCRANCE_FLD=SERVICES_INFO_OCC
FIELDS=DEAL_NAME,SERVICE_TYPE,LOGIN,PASSWD_CLEAR

*FIELD_GROUP
NAME=SERVICES_1
FLIST_NAME=SERVICES
OCCRANCE_FLD=SERVICES_INFO_OCC
FIELDS=LASTSTAT_CMNT,LAST_STATUS_T,STATUS_FLAGS,STATUS,LOGIN,PASS
WD

*FIELD_GROUP
NAME=BILLINFO
FLIST_NAME=BILLINFO
OCCRANCE_FLD=BILLINFO_INFO_OCC
FIELDS=MERCHANT,BILL_MODE,BILL_TYPE,CURRENCY,BILL_WHEN,ACCESS_COD
E1,ACCESS_CODE2

*FIELD_GROUP
NAME=BILLINFO_1
FLIST_NAME=BILLINFO
OCCRANCE_FLD=BILLINFO_INFO_OCC
FIELDS=MERCHANT,BILL_MODE,BILL_TYPE,CURRENCY,BILL_WHEN,ACCESS_COD
E1,ACCESS_CODE2,LAST_BILL_T,NEXT_BILL_T
```

```
*FIELD_GROUP
NAME=BILLINFO_2
FLIST_NAME=BILLINFO
OCCRANCE_FLD=BILLINFO_INFO_OCC
FIELDS=BILL_TYPE,CURRENCY,BILL_WHEN,ACCESS_CODE1,ACCESS_CODE2

*FIELD_GROUP
NAME=ACTGINFO
FLIST_NAME=ACTGINFO
OCCRANCE_FLD=ACTGINFO_INFO_OCC
FIELDS=ACTG_FUTURE_DOM,ACTG_TYPE

*FIELD_GROUP
NAME=ACTGINFO_1
FLIST_NAME=ACTGINFO
OCCRANCE_FLD=ACTGINFO_INFO_OCC
FIELDS=ACTG_CYCLE_DOM,ACTG_TYPE

*FIELD_GROUP
NAME=PAYINFO
FLIST_NAME=PAYINFO
OCCRANCE_FLD=PAYINFO_INFO_OCC
FIELDS=NAME,OBJ_TYPE,INV_INFO,CC_INFO

*FIELD_GROUP
NAME=INV_INFO
FLIST_NAME=INV_INFO
OCCRANCE_FLD=INV_INFO_OCC
FIELDS=NAME,ADDRESS,CITY,STATE,ZIP,COUNTRY,EMAIL_ADDR,DELIVERY_PR
EFER,DELIVERY_DESCR,INV_TERMS,INV_INSTR,PO_TYPE,PO_OPTIONS,PO_AMO
UNT,PO_BAL_THRESHOLD,PO_EXP,PO_EXP_THRESHOLD,PO_ORDER_NO,PO_TERMS

*FIELD_GROUP
NAME=CC_INFO
FLIST_NAME=CC_INFO
OCCRANCE_FLD=CC_INFO_OCC
FIELDS=NAME,ADDRESS,CITY,STATE,ZIP,COUNTRY,DEBIT_NUM,DEBIT_EXP

*FIELD_GROUP
NAME=NAMEINFO
FLIST_NAME=NAMEINFO
OCCRANCE_FLD=NAMEINFO_INFO_OCC
FIELDS=ELEMENT_ID,CONTACT_TYPE,SALUTATION,LAST_NAME,FIRST_NAME,MI
DDLE_NAME,TITLE,COMPANY,ADDRESS,CITY,STATE,ZIP,COUNTRY,EMAIL_ADDR
,PHONES

*FIELD_GROUP
NAME=PHONES
FLIST_NAME=PHONES
```

```
OCCRANCE_FLD=PHONES_INFO_OCC
FIELDS=TYPE,PHONE

*FIELD_GROUP
NAME=EXEMPTIONS
FLIST_NAME=EXEMPTIONS
OCCRANCE_FLD=EXEMPTIONS_INFO_OCC
FIELDS=TYPE,PERCENT

*FIELD_GROUP
NAME=LOCALES
FLIST_NAME=LOCALES
OCCRANCE_FLD=LOCALE_INFO_OCC
FIELDS=LOCALE

*FIELD_GROUP
NAME=BALANCES
FLIST_NAME=BALANCES
OCCRANCE_FLD=BALANCES_INFO_OCC>
FIELDS=ELEMENT_ID,CREDIT_FLOOR,CREDIT_LIMIT,CREDIT_THRESHOLDS,CUR
RENT_BAL,FLAGS,OPEN_BAL

*FIELD_GROUP
NAME=PRODUCTS
FLIST_NAME=PRODUCTS
OCCRANCE_FLD=PRODUCTS_INFO_OCC
FIELDS=ELEMENT_ID,CREATED_T,CYCLE_DISCOUNT,CYCLE_DISC_AMT,CYCLE_E
ND_T,CYCLE_START_T,FLAGS,MMC_END_T,MMC_START_T,MMC_TYPE,PURCHASE_
DISCOUNT,PURCHASE_DISC_AMT,PURCHASE_END_T,PURCHASE_START_T,SERVIC
E_TYPE,LOGIN,PASSWD

*FIELD_GROUP
NAME=PRODUCTS_1
FLIST_NAME=PRODUCTS
OCCRANCE_FLD=PRODUCTS_INFO_OCC
FIELDS=ELEMENT_ID,NAME,DESCR,STATUS,CREATED_T,MOD_T,END_T,OWN_MAX
,OWN_MIN,PARTIAL,PERMITTED,PROVISIONING_TAG,PURCHASE_MAX,PURCHASE
_MIN,START_T,TYPE,STATUS

*FIELD_GROUP
NAME=BILLS
FLIST_NAME=BILLS
OCCRANCE_FLD=BILLS_INFO_OCC
FIELDS=BILL_NO,START_T,END_T,STATUS

*FIELD_GROUP
NAME=BILLS_1
FLIST_NAME=BILLS
OCCRANCE_FLD=BILLS_INFO_OCC
```

```
FIELDS=BILL_NO,START_T,END_T,TOTAL_DUE,CURRENT_TOTAL,CURRENCY

*FIELD_GROUP
NAME=BILLS_2
FLIST_NAME=BILLS
OCCRANCE_FLD=BILLS_INFO_OCC
FIELDS=BILL_NO

*FIELD_GROUP
NAME=BILLS_3
FLIST_NAME=BILLS
OCCRANCE_FLD=BILLS_INFO_OCC
FIELDS=BILL_NO,BUFFER

*FIELD_GROUP
NAME=ITEMS
FLIST_NAME=ITEMS
OCCRANCE_FLD=ITEMS_INFO_OCC
FIELDS=ITEM_NO,NAME,ITEM_TOTAL,CURRENCY,DISPUTED,RECVD,ADJUSTED,D
UE,STATUS

*FIELD_GROUP
NAME=LIMIT
FLIST_NAME=LIMIT
OCCRANCE_FLD=LIMIT_INFO_OCC
FIELDS=ELEMENT_ID,CREDIT_LIMIT,CREDIT_FLOOR,CREDIT_THRESHOLDS

*FIELD_GROUP
NAME=RATES
FLIST_NAME=RATES
OCCRANCE_FLD=RATES_INFO_OCC
FIELDS=CURRENCY,DESCR,CREATED_T,END_T,MOD_T,NAME,PRIORITY,PRORATE
_FIRST,PRORATE_LAST,PURCHASE_MAX,PURCHASE_MIN,RELATIVE_END_T,RELA
TIVE_START_T,VALID_DOM,VALID_DOW,VALID_MOY,TAX_CODE,TAX_WHEN,TYPE
,ZONE,BAL_IMPACTS

*FIELD_GROUP
NAME=BAL_IMPACTS
FLIST_NAME=BAL_IMPACTS
OCCRANCE_FLD=BAL_IMPACTS_INFO_OCC
FIELDS=ELEMENT_ID,FIXED_OPERAND,FLAGS,FREE_QUANTITY,GL_ID,IMPACT_
CATEGORY,SCALED_OPERAND

*FIELD_GROUP
NAME=GETAUTHINFO_INPUT
FLIST_NAME=GETAUTHINFO_INPUT
OCCRANCE_FLD=GETAUTHINFO_OCC
FIELDS=SERVICE_TYPE,LOGIN,PASSWD_CLEAR
```

```
*FIELD_GROUP
NAME=GETAUTHINFO_OUTPUT
FLIST_NAME=GETAUTHINFO_OUTPUT
OCCRANCE_FLD=GETAUTHINFO_OCC
FIELDS=RESULT,TYPE,DESCR

*FIELD_GROUP
NAME=GET_ACCOUNT_NO_INPUT
FLIST_NAME=GET_ACCOUNT_NO_INPUT
OCCRANCE_FLD=GET_ACCOUNT_NO_OCC
FIELDS=SERVICE_TYPE,LOGIN

*FIELD_GROUP
NAME=GET_ACCOUNT_NO_OUTPUT
FLIST_NAME=GET_ACCOUNT_NO_OUTPUT
OCCRANCE_FLD=GET_ACCOUNT_NO_OCC
FIELDS=ACCOUNT_NO

*FIELD_GROUP
NAME=POST_ACTIVITY_INPUT
FLIST_NAME=POST_ACTIVITY_INPUT
OCCRANCE_FLD=POST_ACTIVITY_OCC
FIELDS=LOGIN,SERVICE_TYPE,DESCR,RATES2

*FIELD_GROUP
NAME=POST_ACTIVITY_OUTPUT
FLIST_NAME=POST_ACTIVITY_OUTPUT
OCCRANCE_FLD=POST_ACTIVITY_OCC
FIELDS=OPS_ERROR,DESCR

*FIELD_GROUP
NAME=RATES2
FLIST_NAME=RATES
OCCRANCE_FLD=RATES2_INFO_OCC
FIELDS=NAME,IMPACT_CATEGORY,QUANTITY

*FIELD_GROUP
NAME=LOGIN_INPUT
FLIST_NAME=LOGIN_INPUT
OCCRANCE_FLD=LOGIN_OCC
FIELDS=LOGIN,PASSWD_CLEAR,SERVICE_TYPE,DESCR,START_T

*FIELD_GROUP
NAME=LOGIN_OUTPUT
FLIST_NAME=LOGIN_OUTPUT
OCCRANCE_FLD=LOGIN_OCC
FIELDS=RESULT,TYPE,DESCR,HANDLE
```

```
*FIELD_GROUP
NAME=LOGOUT_INPUT
FLIST_NAME=LOGOUT_INPUT
OCCRANCE_FLD=LOGOUT_OCC
FIELDS=HANDLE,DESCR,END_T,RATES2

*FIELD_GROUP
NAME=LOGOUT_OUTPUT
FLIST_NAME=LOGOUT_OUTPUT
OCCRANCE_FLD=LOGOUT_OCC
FIELDS=OPS_ERROR,DESCR

*FIELD_GROUP
NAME=GET_SESSIONS_INPUT
FLIST_NAME=GET_SESSIONS_INPUT
OCCRANCE_FLD=SESSIONS_OCC
FIELDS=SERVICE_TYPE

*FIELD_GROUP
NAME=GET_SESSIONS_OUTPUT
FLIST_NAME=GET_SESSIONS_OUTPUT
OCCRANCE_FLD=SESSIONS_INFO_OCC
FIELDS=SESSIONS_INFO

*FIELD_GROUP
NAME=SESSIONS_INFO
FLIST_NAME=RESULTS
OCCRANCE_FLD=SESSIONS_OCC
FIELDS=HANDLE,START_T,LOGIN,SERVICE_TYPE,ACCOUNT_NO

*FIELD_GROUP
NAME=DEL_CUST_INPUT
FLIST_NAME=DEL_CUST_INPUT
OCCRANCE_FLD=DEL_CUST_OCC
FIELDS=ACCOUNT_NO

*FIELD_GROUP
NAME=DEL_CUST_OUTPUT
FLIST_NAME=DEL_CUST_OUTPUT
OCCRANCE_FLD=DEL_CUST_OCC
FIELDS=OPS_ERROR,DESCR
```

Modifying the Portal Infranet Configuration File

The eLink Adapter for Portal Infranet uses Portal Infranet libraries to connect to the Portal Infranet system. The file `pin.conf` (Unix) `or pin.cnf` (NT) is a configuration file that Portal Infranet libraries use to find the Portal Infranet system. This file must be installed in the application directory of the eLink Adapter for Portal Infranet. The following entries must be modified in the sample `pin.conf` file that is supplied with the adapter.

Entries that you must modify in the `pin.conf (Unix) or pin.cnf (NT)` file	**Definition of each entry**
`nap login_name <login_name>`	specifies the name to use when the adapter requests a connection to the Portal Infranet system.
`nap login_pw <login_pw>`	specifies the password to use when the adapter requests a connection to the Portal Infranet system.
`nap cm_name <name>`	the host name or TCP/IP address of the Portal Infranet system
`nap cm_port <port>`	the TCP/IP port to connect to the Portal Infranet system

Listing 3-4 is a sample of the supplied example `pin.conf` or `pin.cnf`. The entries that need to be modified are highlighted in bold at the end of the file sample.

Listing 3-4 Sample `pin.conf` (`pin.cnf`)File

```
#====================================================================
# Configuration File for the Sample Object Creation Application
#====================================================================
#
```

```
#====================================================================
# Use this file to specify how the sample application connects with
# Infranet.
#
# A copy of this configuration file is automatically installed and
# configured with default values when you install Infranet. However,
# you can edit this file to suit your specific configuration:
# -- You can change the default value of an entry.
# -- You can exclude an optional entry by adding the # symbol
# at the beginning of the line.
# -- You can include a commented entry by removing the # symbol.
#
# Before you make any changes to this file, save a backup copy.
#
# To edit this file, follow the instructions in the commented sections.
# For more information on the general syntax of configuration entries,
# see "Reference Guide to Infranet Configuration Files" in the Infranet
# online documentation.
#====================================================================
#
#
#====================================================================
# cm_ptr
#
# Pointer to the CM/CMMP.
#
# Use a separate entry for each CM/CMMP. If the application can't
# find the first CM/CMMP, it looks for the next in the list.
#
# Each entry includes three values:
# -- <protocol>: the protocol must be "ip" for this release
# -- <hostname>: name or IP address of the computer running
# the CM/CMMP
# -- <port>: TCP port number of the CM/CMMP on this computer
#
# The port number should match a corresponding cm_ports entry with
# the same port number in the CM/CMMP configuration file. The
# default, 11960, is a commonly specified port for the CM/CMMP.
#====================================================================
- nap cm_ptr ip XXX 11960
#- nap cm_ptr ip <hostname2> <port2>

#====================================================================
# userid
#
# Database number and service type for the Infranet database.
#
# The CM uses the database number to identify the Infranet database
# and connect to the correct Data Manager. For connections that don't
```

```
# require a login name and password, the CM also passes the service
# type to the database.
#
# The database number, in the form 0.0.0.N, is the number assigned to
# your Infranet database when you installed the system. The default
# is 0.0.0.1.
#
# The service type is "/service/pcm_client", with an ID of "1".
# Do not change these values.
#=========================================================================
- - userid 0.0.0.1 /service/pcm_client 1
#=========================================================================
# These entries contain the login information for connecting
# to the CM:
#
# -- "login_type" specifies whether the login name and password
# are required. Type 1, the default, indicates that name and
# password are required. Type 0 indicates that only the user
# ID is required.
#
# -- "login_name" specifies the login name to use when the
# sample application requests a connection to the CM. If
# "login_type" is 0, you don't need to specify a login name.
#
# -- "login_pw" specifies the password to use when the sample
# application connects to the CM. If "login_type" is 0, you
# don't need to specify a password.
#=========================================================================
- nap login_type 1
- nap login_name root
- nap login_pw password
- nap cm_name mynode
- nap cm_port 11960
```

The Portal Infranet Document *Reference Guide to Infranet Configuration Files* contains additional explanation of this configuration file.

4 Invoking Portal Infranet Services

The eLink Adapter for Portal Infranet advertises services that represent operations to be performed on business objects executing in the Portal Infranet billing system. These services allow access to account and billing information that is maintained by Portal Infranet. In order to invoke these services. eLink Platform clients create FML32 buffers and invoke the service of interest with tpcall() or tpacall(). The FML fields that must be supplied for each service are documented in Appendix A, “Portal Infranet Service Interfaces” of this User Guide.

This section describes how to create request buffers for the Portal Infranet services advertised by the adapter. It also describes how to interpret the responses received from these service invocations and the enumerated fields, date/time fields, and the hierarchical nature of Portal Infranet messages.

This chapter contains the following topics:

- Enumerations
- Date/Time Values
- Hierarchical Data
- Creating a Request Buffer
- Traversing a Response Buffer

Enumerations

Certain fields that are sent to or received from the eLink Adapter for Portal Infranet contain enumerated data. These enumerations are documented in Appendix B, “Portal Infranet Enumerations” of this User Guide and are supplied in a C include file `prtconst.h` with the eLink Adapter for Portal Infranet. These enumerated values must be specified by eLink Platform clients for the FML fields that are listed in Appendix A as an enumeration field.

For example, the Create Account interface descried in Appendix A, “Portal Infranet Service Interfaces” of this User Guide denotes a field named DELIVERY_PREFER. This field is listed as a type of ‘enumeration.’ In Appendix B, “Portal Infranet Enumerations” of this User Guide, the numerated values for DELIVERY_PREFER are listed as shown in listing 4-1.

Listing 4-1 Example of a Enumerated Value

```
DELIVERY_PREFER     PIN_INV_EMAIL_DELIVERY = 0
                    PIN_INV_USP_DELIVERY = 1
                    PIN_INV_FAX_DELIVERY = 2
```

When a eLink Platform client wishes to invoke the Create Account service, the FML32 buffer that is created needs to populate the field DELIVERY_PREFER with one of the enumerated values (0, 1, or2) as documented in Appendix B, “Portal Infranet Enumerations”. The sample applications provided with the eLink Adapter for Portal Infranet also illustrate this concept.

Date/Time Values

For fields that are defined as TIMESTAMP in Appendix B, “Portal Infranet Enumerations” of this User Guide, the string for this FML field must be of the correct format. Timestamp values are represented in FML as strings in the following format ‘mm/dd/yyyy HH:MM:SS”. An explanation of these formats appears in Table 4-1.

Format	Description
mm	Two-digit months
dd	Two-digit day of the month
yyyy	Four-digit year
HH	Two-digit hour
MM	Two-digit minute
SS	Two-digit seconds

For example, 09/11/199913:01:25 represents September 11, 1999 1:01:25 p.m. In Appendix A, “Portal Infranet Service Interfaces” of this User Guide, the Create Account interface defines a field LAST_STATUS_T as a TIMESTAMP filed. When this field is received in a response from the Create Account service, it will be a string representing the date and time the account was last modified. This string will be of the format “mm/dd/yyyy HH:MM:SS.”

Hierarchical Data

The major difficulty in using Portal business objects is that the messages that can be sent to or received from the Portal Infranet system are hierarchical in nature. Applications within the Portal Infranet system communicate using hierarchical

messages that can have nested structures, and arrays are embedded at any level. The eLink Platform system does not have a native hierarchical buffer type that allows structures to be embedded within structures.

The eLink Adapter for Portal Infranet has adopted a scheme of flattening hierarchical structured data into a tag/value FML32 buffer. An implicit order is used to retain the hierarchical order. Examples of this concept are included with the eLink Adapter for Portal Infranet that further illustrate how messages must be passed to the adapter and how to interpret responses received from Portal Infranet.

Listing 4-2 illustrates a message that can be sent to the Portal Infranet system in order to create a customer account. Each level of indentation represents a level in the hierarchical structure.

The Create Account message at the highest level consists of an EA_PLAN_NAME filed, an array of SERVICES and an array of PAYINFO. A SERVICE structure is composed of a DEAL_NAME, SERVICE_TYPE, LOGIN, and PASSWD_CLEAR field. The PAYINFO structure consists of a NAME field, an OBJ_TYPE field, an array of INV_INFO (invoice information) and an array of CC_INFO (credit card information).

Listing 4-2 Creates a Customer Account Message in Portal Infranet

```
Field EA_PLAN_NAME
Array  SERVICES
       Fields
             DEAL_NAME
             SERVICE_TYPE
             LOGIN
             PASSWD_CLEAR
Array  PAYINFO
       Fields
             NAME
             OBJ_TYPE
             Array  INV_INFO
                    Fields
                          NAME
                          ADDRESS
                          CITY
                          STATE
                          ZIP
                          COUNTRY
                          EMAIL_ADDR
                          DELIVERY_PREFER
```

```
                    DELIVERY_DESCR
                    INV_TERMS
                    INV_INSTR
                    PO_TYPE
                    PO_OPTIONS
                    PO_AMOUNT
                    PO_BAL_THRESHOLD
                    PO_EXP
                    PO_EXP_THRESHOLD
                    PO_ORDER_NO
                    PO_TERMS
             Array CC_INFO
                   Fields
                    NAME
                    ADDRESS
                    CITY
                    STATE
                    ZIP
                    COUNTRY
                    DEBIT_NUM
                    DEBIT_EXP
```

An FML32 buffer is a tag/value structured buffer of data that only maintains the fields that are in the buffer and the number of occurrences of each field exist in the buffer. Grouping of fields, the fact that multiple fields are part of a structure, is not maintained. In order to represent structured data in an FML32 buffer, two constraints are used to maintain the hierarchical groupings of fields:

1. Fields must be added to an FML 32 buffer in pre-order traversal. Fields must be added to an FML32 buffer in the order listed for the service in Appendix A, “Portal Infranet Service Interfaces” of this User Guide. Substructures must be traversed, and their fields must be added as listed. In the example above, the fields must be added to the FML32 buffer in the following order:
   - PLAN_NAME
   - DEAL_NAME
   - SERVICE_TYPE
   - LOGIN
   - PASSWD_CLEAR
   - NAME

- OBJ_TYPE
- NAME
- ADDRESS

The first occurrence of the NAME field relates to the PAYINFO structure, and the next occurrence of NAME should be associated with the INVINFO structure. This relationship can only be maintained by agreeing upon the order that the fields are 'flattened' into an FML32 buffer.

2. The number of occurrences of each group must be explicitly specified in the FML32 buffer. For each 'group' of fields, an extra FML32 field must be added to the data to denote how many occurrences of this group exist in the FML32 data. In the example above, an extra field SERVICES_OCC would indicate how many occurrences of the SERVICES information exist in the FML32 buffer.

Creating a Request Buffer

The constraints described in the Enumerations, Date/Time Values, and Hierarchical Data sections of this chapter are best illustrated in an example. The following example provides pseudocode to create an FML32 buffer to request the Create Account service. This sample client, shown below, is provided with the eLink Adapter for Portal Infranet, and it also illustrates how to create an FML32 buffer for the Create Account service. To create an FML32 buffer for a Create Account service, follow these instructions:

1. Allocate and FML32 buffer using tpalloc().
2. Initialize FML32 buffer using Finit32().
3. Add PLAN_NAME FIELD for this account.
4. Add SERVICES_OCC field to indicate how many services were subscribed by this account.
5. Add DEAL_NAME, SERVICE_TYPE, LOGIN, PASSWD_CLEAR fields for each service.
6. Add PAYINFO_OCC field to indicate how many payment info records are associated with this account.

7. Add NAME and OBJ_TYPE fields for payment information.
8. Add INV_INFO_OCC field to indicate how many invoice info records are associated with this payment information.
9. Add NAME, ADDRESS, CITY, STATE, ZIP, COUNTRY, EMAIL_ADDR for each invoice information record.

The resulting FML 32 buffer is shown in Listing 4-3.

Listing 4-3 FML32 Request Buffer for "Create Account"

```
PLAN_NAME[0]          "ISP Plan"
SERVICES_OCC[0]       1
DEAL_NAME[0]          "ISP-1"
SERVICE_TYPE[0]       "Internet Dial-up"
LOGIN[0]              "jdoe"
PASSWD_CLEAR[0]       "abcdefg"
PAYINFO_OCC[0]        1
NAME[0]               "Credit account"
OBJ_TYPE[0]           "Payment"
INV_INFO_OCC[0]       1
NAME[1]               "John Doe"
ADDRESS[0]            "101 Doe Ln"
CITY[0]               Denver
STATE[0]              Colorado
ZIP[0]                99999
COUNTRY[0]            United States
EMAIL_ADDR[0]         jdoe@email.org
```

Traversing a Response Buffer

After a request is sent to a service advertised by the eLink Adapter for Portal Infranet, the response buffer is returned. This response message (depending on the invoked service) can contain hierarchical data. The information that follows demonstrates how you can retrieve this data and the implied hierarchical structure of that data. Listing 4-4 shows an example of the response buffer for Create Account.

Listing 4-4 Response Buffer for Create Account

```
ACCOUNT NO[0]        123456789
SERVICES_OCC[0]      1
LASTSTAT_CMNT[0]     “Account Created”
LAST STATUS[0]
STATUS FLAGS[0]      0
STATUS[0]            10100
LOGIN[0]             “jdoe”
PASSWD_CLEAR[0]      “abcdef1”
PAYINFO_OCC[0]       1
NAME[0]              “Credit account”
OBJ_TYPE             “Payment”
INV_INFO_OCC[0]      1
NAME[0]              “John Doe”
ADDRESS[0]           “101 Doe Ln”
CITY[0]              Denver
STATE[0]             Colorado
ZIP[0]               99999
COUNTRY[0]           United States
EMAIL_ADDR[0]        jdoe@email.org
```

The interface description in Appendix B, “Portal Infranet Enumerations” of this User Guide lists the fields that are present in a Create Account response message. To retrieve this data from the FML buffer, it is important to know which occurrence of a field belongs to which group (payment info, invoice info, etc.) The following algorithm retrieves data from the FML buffer and deletes each field as it is retrieved. This algorithm does not keep track of the occurrence of an FML field. Successive occurrences are shifted upward, and then they become first occurrence (NAME[0]). This algorithm is also demonstrated in the supplied sample applications. To traverse a response buffer, follow these instructions:

1. Use Fget32() to retrieve ACCOUNT_NO data.
2. Use Fdel32() to delete this occurrence of ACCOUNT_NO data.
3. Use Fget32() to retrieve SERVICES_OCC data.
4. Use Fdel32() to delete this occurrence of the field SERVICES_OCC.
    a. For all occurrences of SERVICES (the value retrieved in SERVICES_OCC)

   b. For all fields of the SERVICES group (LASTSTAT_CMNT, LAST_STATUS_T, STATUS_FLAGS, SATATUS, LOGIN, PASSWD_CLEAR)

   c. Use Fget32() to retrieve the data.

   d. Use Fdel32() to delete the occurrence of this field.

5. Use Fget32() to retrieve PAYINFO_OCC data.
6. Use Fdel32() to delete this occurrence of the field PAYINFO_OCC.

   a. For all occurrences of PAYMENT INFO (the value retrieved in PAYINFO_OCC).

7. Use Fget32() to retrieve the NAME field.
8. Use Fdel32() to delete this occurrence of the NAME field.
9. Use Fget32() to retrieve the OBJ_TYPE field.
10. Use Fdel32() to delete this occurrence of the OBJ_TYPE field.
11. Use Fget32() to retrieve INV_INFO_OCC data.
12. Use Fdel32() to delete this occurrence of the field INV_INFO_OCC).

    a. For all fields of the INVOICE INFO group (NAME, ADDRESS, CITY, STATE, ZIP, COUNTRY, EMAIL_ADDR).

    b. Use Fget32() to retrieve the data.

    c. Use Fdel32() to delete this occurrence of this field.

5 Running the Sample Application

The information in this section is designed to help you start and run a sample application. This section contains the following information:

- Step 1: Copy the simpportal Files
- Step 2: Set Up Environment Files
- Step 3: Edit the eLink Platform Configuration File
- Step 4: Load the eLink Platform Configuration File
- Step 5: Modify Portal Infranet Configuration File (pin.conf or pin.cnf)
- Step 6: Boot the Application
- Step 7: Create a Customer Account
- Step 8: Shutdown the Application

Step 1: Copy the simpportal Files

Perform two primary tasks to copy the `simpportal` files:

- Make a directory for `simpportal`
- Copy the `simpportal` files

Step 1.1: Make a Directory for simpportal

Make a directory for the `simpportal` and `cd` to it so you can see clearly the `simpportal` files you have at the start and the additional files you create along the way. Use the standard shell (`/bin/sh`) or the Korn shell, not `csh`.

To make a directory, type the following:

```
mkdir simpportal
cd simpportal
```

Step 1.2: Copy the simpportal Files

Later in the process you will edit some of the files and make them executable, so it is best to begin with a copy of the files rather than the originals delivered with the software. The directory shown in Listing 5-1 contains sample files for configuring the eLink Adapter for Portal Infranet. To copy the `simpportal` files, type the following:

```
cp $TUXDIR/elink/portal/simpportal ./
```

Listing 5-1 Description of Sample Files

File Name	Description
`elinkportal.ubb`	sample eLink Platform `ubbconfig` file
`elinkportal.env`	sample eLink Platform ENVFILE for eLink for Portal Infranet adapter
`elinkportal.cfg`	sample eLink for Portal Infranet adapter configuration file
`pin.conf` (Unix) or `pin.cnf` (NT)	sample Portal Infranet configuration file
`setenv.sh`	sample UNIX Korn Shell script to set environment
`create_cust.c`	sample eLink Platform client application
`prtconst.h`	header file that defines constants for Portal Infranet data
`setenv.bat`	sample NT script to set the environment

Step 2: Set Up Environment Files

The information in this section is designed to help you perform the following two setup tasks on both platforms:

- Setting up and exporting environment variables for the Unix platform
- Editing the adapter environment for the Unix platform

Step 2.1A Set and Export Environment Variables for the Unix Platform

For the Unix platform to set and export the environment variables, edit the `setenv.sh` file. You need the `TUXDIR` and `PATH` to access files in the eLink Platform System/T directory structure and execute eLink Platform System/T commands. With HPUX on the HP9000, use `SHLIB_PATH` instead of `LD_LIBRARY_PATH`. Listing 5-2 shows a sample of the environment variables for the Unix platform.

Note: You must set up the `TUXCONFIG` to be able to load the configuration file.

To set and export the environment variables for a Unix platform, edit the provided file `setenv.sh`. Using a standard text editor, follow these steps:

1. Bring up `setenv.sh` in your text editor.
2. Replace the fields delimited with '<' and '>' signs.
3. Set the shared library path for Unix:
   ```
   SHLIB_PATH=$(TUXDIR)/lib:$(SHLIB_PATH)
   ```
4. Set the environment by executing this script:
   ```
   . . /setenv.sh
   ```

Listing 5-2 Sample Environment Variables File for the Unix Platform

```
#!/bin/sh
APPDIR=<your eLink app directory>
export APPDIR
TUXDIR=<your TUXEDO install directory>
export TUXDIR
PINDIR=<your Portal Infranet install directory>
export PINDIR
TUXCONFIG=${APPDIR}/tuxconfig
export TUXCONFIG
BDMCONFIG=${APPDIR}/bdmconfig
export BDMCONFIG
FIELDTBLS32=elinkportal.fml,Usysfl32
export FIELDTBLS32
FLDTBLDIR32=${APPDIR}:${TUXDIR}/udataobj
export FLDTBLDIR32
PATH=${TUXDIR}/bin:${PATH}
# ------------------------------------------
# Set shared library path according to platform
# ------------------------------------------
#
# For HP-UX use SHLIB_PATH
#
#SHLIB_PATH=${TUXDIR}/lib:${PINDIR}/lib:${SHLIB_PATH}
#export SHLIB_PATH
#
# For Solaris use LD_LIBRARY_PATH
#
#LD_LIBRARY_PATH=${TUXDIR}/lib:${PINDIR}/lib:${LD_LIBRARY_PATH}
#export LD_LIBRARY_PATH
#
# For AIX use LIBPATH
#
#LIBPATH=${TUXDIR}/lib:${PINDIR}/lib:${LIBPATH}
#export LIBPATH
```

Note: `PINDIR` should point to the directory where Portal Infranet is installed.

Step 2.1B Set and Export Environment Variables for the NT Platform

To set and export the environment variables, you must edit the `setenv.bat` file if you are using a NT 4.0 platform. You need the `TUXDIR` and `PATH` to access files in the eLink Platform System/T directory structure and execute eLink Platform System/T commands. Listing 5-3 shows a sample of the environment variables for the NT 4.0 platform.

Note: You must set up the `TUXCONFIG` to be able to load the configuration file.

To set and export the environment variables for a NT platform, edit the provided file `setenv.bat`. Using a standard text editor, follow these steps:

1. Bring up `setenv.bat` in your text editor.
2. Replace the fields delimited with '<' and '>' signs.
3. Set the environment by executing this script:

```
setenv
```

Listing 5-3 Sample Environment Variables File for the NT Platform

```
set APPDIR=<Your eLink app directory>
set TUXDIR=<Your TUXEDO install directory>
set PINDIR=<Your Portal Infranet install directory>
set TUXCONFIG=%APPDIR%\tuxconfig
set BDMCONFIG=%APPDIR%\bdmconfig
set FIELDTBLS32=elinkportal.fml,Usysfl32
set FLDTBLDIR32=%APPDIR%;%TUXDIR%\udataobj
set PATH=%TUXDIR%\lib;%PINDIR%\bin;%PATH%
set LIB=%TUXDIR%\lib;%PINDIR%\lib;%LIB%
```

Note: `PINDIR` should point to the directory where Portal Infranet is installed.

Step 2.2 Edit the Adapter Environment File

You must edit the adapter environment file to allow the adapter to access system and user-defined FML field tables. Table 5-1 provides a description of the two environment variables to access FML tables. Listing 5-4 shows a sample adapter environment file for the Unix platform, and Listing 5-5 shows the file for an NT platform.

Using a standard text editor, follow these steps:

1. Access the environment file `elinkportal.env`.
2. Replace the fields delimited with '<' and '>' signs.

Table 5-1 Environment Variables to Access FML Tables

Variable Name	Description
FIELDTBLS32	Lists the FML field tables that are to be access by the eLink Adapter for Portal Infranet.
FLDTBLDIR32	Lists the directories to be searched for these FML field tables.

Listing 5-4 Sample Adapter Environment File for a Unix Platform

```
FIELDTBLS32=Usysfl32,elinkportal.fml

FLDTBLDIR32=<your eLink app directory>:<your TUXEDO install
directory>/udataobj
```

Listing 5-5 Sample Adapter Environment File for an NT Platform

```
FIELDTBLS32=Usysfl32,elinkportal.fml

FLDTBLDIR32=<your eLink app directory>;<your TUXEDO install
directory>/udataobj
```

Step 3: Edit the eLink Platform Configuration File

To define the ELINKPORTALO server, add the ELINKPORTALO information in the SERVERS section of the UBBCONFIG file. The following parameters are required for defining the ELINKPORTALO server. Listing 5-6 shows the parameters for defining the server.

Listing 5-6 Parameters for Defining the ELINKPORTALO Server

```
*SERVERS

ELINKPORTALO
       SRVGRP=groupname     SRVID=n
       CLOPT="-- -C elinkportal.cfg"
```

For information about the SRVGRP, SRVID, and CLOPT parameter syntax and definitions, refer to the *BEA TUXEDO Reference Manual.*

CLOPT= "-- -C *configfile*"
specifies the adapter's configuration file.

Listing 5-7 shows a sample configuration file.

Listing 5-7 Sample Configuration File

```
#####################################################################
# This is a skeletal TUXEDO configuration file - "ePS.ubb" designed
# to be used for BEA eLink Adapter for ELINKPORTALO.ubb
#
#####################################################################
# @(#)$Header: /repos/port/sample/simpportal/elinkportal.ubb,v 1.1
1999/10/04 22:27:18 schupbac Exp $
# Copyright )1999, BEA Systems, Inc., all rights reserved.

*RESOURCES
```

```
IPCKEY          33248        # ( 32768 < IPCKEY < 262143 )
MASTER          eLink
DOMAINID        eLink
MODEL           SHM
MAXSERVERS      20
SECURITY        NONE
#SECURITY       USER_AUTH

*MACHINES
"<uname>" LMID="eLink"
        TUXDIR="<your TUXEDO install directory>"
        APPDIR="<your eLink app directory>"
        TUXCONFIG="<your eLink app directory>/tuxconfig"
        ENVFILE="<your eLink app directory>/elinkportal.env"
        TYPE="HP-UX"
        MAXACCESSERS=80

*GROUPS
EPORTALGRPLMID=eLink

       GRPNO=1


#
------------------------------------------------------------------
-----

*SERVERS

DEFAULT:
       CLOPT="-A"       # Advertise all services.
       REPLYQ=N       # Reply queue not needed for our simple setup.
        MAXGEN=3       # Max number of restarts in the grace period.
       GRACE=60         # Ten minutes grace period.
       RESTART=Y
        SYSTEM_ACCESS=FASTPATH

ELINKPORTALO
              SRVGRP=EPORTALGRP
                SRVID=200
                CLOPT="-A -- -C elinkportal.cfg"

*SERVICES
```

Step 4: Load the eLink Platform Configuration File

Perform two primary tasks to load the eLink Platform configuration file:

- Run `tmloadcf` to load the file.
- Check the configuration file to ensure that it is called `tuxconfig`.

Step 4.1: Load the File

Run `tmloadcf` to load the configuration file.

```
$ tmloadcf elinkportal.ubb
Initialize TUXCONFIG file:  /usr/me/simpportal/tuxconfig [y, q] ? y
$
```

Step 4.2: Check the Results

Check to ensure that a file called `tuxconfig` is a new file under the control of eLink Platform System/T. Listing 5-8 shows a sample eLink Platform configuration file.

Listing 5-8 Sample eLink Platform Configuration File

```
$ ls -l tuxconfig
total 216
-rw-r-----   1 userid    grpid    106496  May  29  09:26  tuxconfig
```

Step 5: Modify Portal Infranet Configuration File (pin.conf or pin.cnf)

The eLink Adapter for Portal Infranet uses Portal Infranet libraries to connect to the Portal Infranet system. The file `pin.conf` (Unix) `pin.cnf` (NT) is a configuration file that Portal Infranet libraries use to find the Portal Infranet system. This file must be installed in the application directory of the eLink Adapter for Portal Infranet. The following entries must be modified in the sample `pin.conf` (or `pin.cnf`) file that is supplied with the adapter

Examine the sample Portal Infranet configuration file shown in Listing 3-4 in Chapter 3, “Configuring the eLink Adapter for Portal Infranet,”of this manual. This configuration connects the adapter to a Portal Infranet system on node mynode port 11960 using a login of 'root' and a password of 'password'.

Step 6: Boot the Application

Executing `tmboot` brings up the adapter. Listing 5-9 shows a sample boot process. To execute `tmboot`, type the following:

```
tmboot -y.
```

Listing 5-9 Sample Boot Process

```
INFO: TUXEDO(r) System Release 6.5
INFO: Serial #: 1000039580, Expiration 2000-02-16, Maxusers 10000
INFO: Licensed to: FOO SYSTEMS

Booting admin processes ...

exec BBL -A :
       process id=20993 ... Started.

Booting server processes ...
```

```
exec ELINKPORTALO -A -- -C elinkportal.cfg :
       process id=20994 ... Started.
2 processes started.
```

BBL is the administrative process that monitors the application shared memory structures. The server (`ELINKPORTALO`) runs continuously awaiting requests.

Step 7: Create a Customer Account

Included with the eLink Adapter for Portal Infranet is a sample eLink Platform client program. This supplied client create_cust.c creates a customer named John W. Doe within the Portal Infranet system. This client takes the service name as a parameter and creates the customer John W. Doe from hard-coded test data. The client creates an FML32 request buffer containing the hierarchical data for the new customer John W. Doe. This buffer is then sent to the Portal Infranet adapter service, and the response is returned. The FML32 response buffer is printed to standard out using the FML function Fprint32().

Step 7.1 Examine the Sample Client Program

Before you build the sample client program, examine the program for several important considerations. Table 5-2 provides a key to some important information for you to consider as you examine the sample client program. Listing 5-10 shows the source code for the sample client program. Use the following steps to access the source code:

1. Access the source code by typing: `$ more create_cust.c`
2. Page through the client program and look for the following information:

Table 5-2 Sample Client Program Information Key

Line Number	Significant Information
39-250	Hierarchical data and structures are declared for the John W. Doe customer.
283	The Portal Infranet adapter service is the first command line parameter.
311	add_account_info() is called to populate the FML32 request buffer with hierarchical data for the new customer John W. Doe.
322	Portal Infranet adapter service is invoked.
371	add_account_info() adds the FML32 field ACCOUNT_INFO_OCC to explicitly indicate how many occurrences of account info exist in this buffer.
379-386	For each account that is to be added, information is added to the FML32 buffer that describes what services this account is subscribing too, billing information for this account, etc. A function is used to add each group of data. The CREATE_CUST interface is also documented in the The CREATE ACCOUNT section of Appendix A, “Portal Infranet Service Interfaces” in this manual.
399, 420, 448, 468, 520, 555, 613, 636, 655, 675	Each function that adds a group of data to the FML32 buffer, adds an ‘occurrence’ field to indicate how many times this group occurs in the buffer.

Listing 5-10 Source code for the Sample Client Program

```
1 /*   Copyright 1999 BEA Systems, Inc.*/
2 /*   THIS IS UNPUBLISHED PROPRIETARY SOURCE CODE OF     */
3 /*   BEA Systems, Inc.                        */
4 /*   The copyright notice above does not evidence any   */
5 /*   actual or intended publication of such source code.*/
6
7
8 /*************************************************/
9 /*    test case 1 create_cust.c                  */
10 /*************************************************/
11
12
13 #include <stdio.h>
```

```
14 #include <string.h>
15
16 #include "atmi.h"
17 #include "fml32.h"
18 #include "prtconst.h"
19
20 /* Define constants for creating FML buffer */
21 #define NF      256
22 #define DATALEN 32000
23
24 /*---------------------*/
25 /* Sample data Section */
26 /*---------------------*/
27 #define NUM_ACCOUNT_INFO 1
28 #define NUM_SERVICES 1
29 #define NUM_BILLINFO 1
30 #define NUM_ACTGINFO 1
31 #define NUM_PAYINFO 1
32 #define NUM_INVINFO 1
33 #define NUM_CCINFO  1
34 #define NUM_NAMEINFO 1
35 #define NUM_EXEMPTIONS 1
36 #define NUM_LOCALES 1
37 #define NUM_PHONESINFO 1
38
39 char sample_plan_name[] = "Standard";
40
41 /* Accounting information structure */
42 struct actginfo
43 {
44   long actg_future_dom;
45   long actg_type;
46 };
47
48 struct actginfo sample_actginfo =
49 {
50   18,
51  EPA_ACTG_TYPE_BALANCE_FORWARD
52 };
53
54 /* Structure for services FIELD_GROUP */
55 struct services
56 {
57   char deal_name[BUFSIZ];
58   char service_type[BUFSIZ];
59   char login[BUFSIZ];
60   char passwd_clear[BUFSIZ];
61 };
62
63 struct services sample_services =
64 {
65   "IP Basic",/* should be present valid deal name in portal */
66   "/service/email",/* service already available with portal */
```

```
67   "jdoe",   /* Unique login should not be present in portal database
*/
68   "record"
69 };
70
71 /* billinfo FIELD_GROUP structure */
72 struct billinfo
73 {
74   char merchant[BUFSIZ];
75   char bill_mode[BUFSIZ];
76   long bill_type;
77   long currency;
78   long bill_when;
79   char access_code1[BUFSIZ];
80   char access_code2[BUFSIZ];
81 };
82
83 struct billinfo sample_billinfo =
84 {
85   "Microsoft",
86   "Email",
87   EPA_BILL_TYPE_CC,
88   EPA_CURRENCY_USD,
89   EPA_MMC_TYPE_MONTHLY,
90   "Access Code1",
91   "Access Code2"
92 };
93
94 /* payinfo FIELD_GROUP structure */
95 struct payinfo
96 {
97   char name[BUFSIZ];
98   char obj_type[BUFSIZ];
99 };
100
101 struct payinfo sample_payinfo =
102 {
103   "Payment information",
104   "PAY obj type"
105 };
106
107 /* Invoice info FIELD_GROUP structure */
108 struct invinfo
109 {
110   char name[BUFSIZ];
111   char address[BUFSIZ];
112   char city[BUFSIZ];
113   char state[BUFSIZ];
114   char zip[BUFSIZ];
115   char country[BUFSIZ];
116   char email_addr[BUFSIZ];
117   long  delivery_prefer;
118   char delivery_descr[BUFSIZ];
```

```
119   long  inv_terms;
120   char inv_instr[BUFSIZ];
121   long  po_type;
122   long  po_options;
123   double po_amount;
124   double po_bal_threshold;
125   char po_exp[BUFSIZ];
126   char po_exp_threshold[BUFSIZ];
127   char po_order_no[BUFSIZ];
128   char po_terms[BUFSIZ];
129 };
130
131 struct invinfo sample_invinfo =
132 {
133   "John W. Doe",
134   "123 Park Dr",
135   "Cupertino",
136   "CA",
137   "95014",
138   "USA",
139   "jdoe@beasys.com",
140   EPA_INV_USP_DELIVERY,
141   "Delivery descr on invoice",
142   EPA_INV_TERMS_UNDEFINED,
143   "Invoice instructions on invoice",
144   EPA_PO_TYPE_NONE,
145   EPA_PO_OPTION_CONTINUE,
146   1111.23,
147   1.23
148
149 };
150
151 /* Credit card info FIELD_GROUP strucure */
152 struct ccinfo
153 {
154   char name[BUFSIZ];
155   char address[BUFSIZ];
156   char city[BUFSIZ];
157   char state[BUFSIZ];
158   char zip[BUFSIZ];
159   char country[BUFSIZ];
160   char debit_num[BUFSIZ];
161   char  debit_exp[BUFSIZ];
162  };
163
164 struct ccinfo sample_ccinfo =
165 {
166   "John W. Doe",
167   "123 Main St",
168   "Cupertino",
169   "CA",
170   "95014",
171   "USA",
```

```
172   "4444111122223333",
173   "1201"
174
175 };
176
177
178 /* locales FIELD_GROUP structure */
179 struct locales
180 {
181   char locale[BUFSIZ];
182
183 };
184
185 struct locales sample_locales =
186 {
187   "English (united kingdom) :ENG"
188 };
189
190 /* Phones FIELD_GROUP structure */
191 struct phones
192 {
193    long type;
194    char phone[BUFSIZ];
195 };
196
197 struct phones sample_phones =
198 {
199    EPA_PHONE_TYPE_WORK,
200    "972-555-7130"
201 };
202
203 struct exemptions
204 {
205    long type;
206    double percent;
207 };
208
209 struct exemptions sample_exemptions =
210 {
211    EPA_RATE_TAX_JUR_COUNTY,
212    8.25
213 };
214
215 struct nameinfo
216 {
217   char contacttype[BUFSIZ];
218   char salutation[BUFSIZ];
219   char lastname[BUFSIZ];
220   char firstname[BUFSIZ];
221   char middlename[BUFSIZ];
222   char title[BUFSIZ];
223   char company[BUFSIZ];
224   char address[BUFSIZ];
```

```
225   char city[BUFSIZ];
226   char state[BUFSIZ];
227   char zip[BUFSIZ];
228   char country[BUFSIZ];
229   char email_addr[BUFSIZ];
230   long element_id;
231
232 };
233
234 struct nameinfo sample_nameinfo =
235 {
236   "Residence",
237   "Mr.",
238   "John",
239   "Doe",
240   "W",
241   "software engineer",
242   "BEA Systems",
243   "123 Main St.",
244   "Cupertino",
245   "CA",
246   "95014",
247   "USA",
248   "jdoe@beasys.com",
249   1
250 };
251
252 /*-------------------------*/
253 /* End Sample data Section */
254 /*-------------------------*/
255
256 void add_fld(FBFR32* fbfr, char* fldid, void* value, FLDLEN32 len);
257 void add_account_info(FBFR32 *fbfr, int num_accounts);
258 void add_services(FBFR32 *fbfr, int num_services);
259 void add_billinfo(FBFR32 *fbfr, int num_billinfos);
260 void add_payinfo(FBFR32 *fbfr, int num_payinfos);
261 void add_invinfo(FBFR32 *fbfr, int num_invinfos);
262 void add_ccinfo(FBFR32 *fbfr, int num_ccinfos);
263 void add_nameinfo(FBFR32 *fbfr, int num_nameinfos);
264 void add_phonesinfo(FBFR32 *fbfr, int num_phonesinfos);
265 void add_exemptions(FBFR32 *fbfr, int num_exemptions);
266 void add_actginfo(FBFR32 *fbfr, int num_actginfo);
267 void add_locales(FBFR32 *fbfr, int num_locales);
268
269 int main(int argc, char *argv[])
270 {
271    FBFR32 *fbfr;/* data to be sent */
272         FLDLEN32 fbfr_len;
273    FLDID32 field_id;
274         char *service;
275         char *response;
276    int ret;
277
```

```
278         if (argc != 2)
279         {
280            (void) fprintf(stderr, "Usage: create_cust SERVICE");
281            exit(1);
282         }
283         service=argv[1];
284
285     /* join the application */
286          if (tpinit(NULL) == -1)
287             {
288                    (void) fprintf(stderr,"failed to join
application: %s\n",
289                    tpstrerror(tperrno));
290              exit(1);
291         }
292
293 /*-----------------------------*/
294 /* Create FML32 request buffer */
295 /*-----------------------------*/
296
297     /* get request buffer */
298     if ((fbfr = (FBFR32 *)
tpalloc("FML32",NULL,Fneeded32(NF,DATALEN)))== (FBFR32 *) NULL)
299     {
300            (void) fprintf(stderr,"unable to allocate buffer: %s",
301                    tpstrerror(tperrno));
302            exit(1);
303     }
304
305          fbfr_len = (int) Fsizeof32(fbfr);
306          Finit32(fbfr, (FLDLEN32) fbfr_len);
307
308 /*-------------------------------------------*/
309 /* Build create account FML32 request buffer */
310 /*-------------------------------------------*/
311          add_account_info(fbfr, NUM_ACCOUNT_INFO);
312
313     /*---------------------------*/
314          /* Dump FML32 request buffer */
315     /*---------------------------*/
316          printf("Before:\n\n");
317          Fprint32(fbfr);
318
319 /*--------------*/
320 /* Call service */
321 /*--------------*/
322        ret = tpcall(service, (char *)fbfr, 0, (char **)&fbfr, (long
*) &fbfr_len, (long)0);
323 /*
324          if(ret == -1)
325          {
326               (void) fprintf(stderr, "Can't send request to service
%s\n", service);
```

```
327                (void) fprintf(stderr, "Tperrno = %d\n", tperrno);
328                 tpfree((char *) fbfr);
329                   tpterm();
330                 exit(1);
331         }
332 */
333
334    /*---------------------------*/
335         /* Dump FML32 response buffer */
336    /*---------------------------*/
337         printf("After:\n\n");
338         Fprint32(fbfr);
339
340
341 /*-------------------*/
342 /* Clean up and exit */
343 /*-------------------*/
344    tpfree((char *)fbfr);
345
346    (void) tpterm();
347
348    return(0);
349 }
350
351
352 void add_fld(FBFR32* fbfr, char* fldid, void* value, FLDLEN32 len)
353 {
354         FLDID32 fieldid;
355
356         fieldid = Fldid32(fldid);
357         if (Fadd32(fbfr, fieldid, (char*)value, len) < 0)
358         {
359        char msg[25];
360        sprintf(msg, "Fadd32 %s failed:  ", fldid);
361            F_error32(msg);
362         }
363 }
364
365 void add_account_info(FBFR32 *fbfr, int num_accounts)
366 {
367
368         char temp_string[BUFSIZ];
369    int i;
370
371         add_fld(fbfr, "ACCOUNT_INFO_OCC", (char *) &num_accounts,
372                  (FLDLEN32) sizeof(num_accounts));
373
374    for (i=0; i < num_accounts; i++)
375    {
376
377           add_fld(fbfr, "NAME", sample_plan_name,
(FLDLEN32)sizeof(sample_plan_name));
378
```

```
379       add_services(fbfr, NUM_SERVICES);
380       add_billinfo(fbfr, NUM_BILLINFO);
381       add_actginfo(fbfr, NUM_ACTGINFO);
382       add_payinfo(fbfr, NUM_PAYINFO);
383       add_nameinfo(fbfr, NUM_NAMEINFO);
384       add_phonesinfo(fbfr, NUM_PHONESINFO);
385       add_exemptions(fbfr, NUM_EXEMPTIONS);
386       add_locales(fbfr, NUM_LOCALES);
387
388          }
389
390 }
391
392
393 void add_services(FBFR32 *fbfr, int num_services)
394 {
395
396          char temp_string[BUFSIZ];
397     int i;
398
399         add_fld(fbfr, "SERVICES_INFO_OCC", (char *) &num_services,
(FLDLEN32) sizeof(num_services));
400
401     for (i=0; i < num_services; i++)
402     {
403
404        add_fld(fbfr, "DEAL_NAME", sample_services.deal_name,
(FLDLEN32) sizeof(sample_services.deal_name));
405
406
407       add_fld(fbfr, "SERVICE_TYPE", sample_services.service_type,
(FLDLEN32)sizeof(sample_services.service_type));
408
409        add_fld(fbfr, "LOGIN", sample_services.login, (FLDLEN32)
sizeof(sample_services.login));
410
411       add_fld(fbfr, "PASSWD_CLEAR", sample_services.passwd_clear,
(FLDLEN32) sizeof(sample_services.passwd_clear));
412
413          }
414 }
415 void add_billinfo(FBFR32 *fbfr, int num_billinfos)
416 {
417     char temp_string[BUFSIZ];
418     int i;
419
420     add_fld(fbfr, "BILLINFO_INFO_OCC", (char *) &num_billinfos,
(FLDLEN32) sizeof(num_billinfos));
421
422     for (i=0; i < num_billinfos; i++)
423     {
424
425        add_fld(fbfr, "MERCHANT", sample_billinfo.merchant,
```

```
(FLDLEN32) sizeof(sample_billinfo.merchant));
426
427       sprintf(temp_string, sample_billinfo.bill_mode);
428       add_fld(fbfr, "BILL_MODE", temp_string, (FLDLEN32)
strlen(temp_string));
429
430       add_fld(fbfr, "BILL_TYPE", &sample_billinfo.bill_type,
(FLDLEN32) sizeof(sample_billinfo.bill_type));
431
432       add_fld(fbfr, "CURRENCY", &sample_billinfo.currency,
(FLDLEN32) sizeof(sample_billinfo.currency));
433
434       add_fld(fbfr, "BILL_WHEN", &sample_billinfo.bill_when,
(FLDLEN32) sizeof(sample_billinfo.bill_when));
435
436       sprintf(temp_string, sample_billinfo.access_code1);
437       add_fld(fbfr, "ACCESS_CODE1", temp_string, (FLDLEN32)
strlen(temp_string));
438
439       sprintf(temp_string, sample_billinfo.access_code2);
440       add_fld(fbfr, "ACCESS_CODE2", temp_string, (FLDLEN32)
strlen(temp_string));
441    }
442 }
443 void add_payinfo(FBFR32 *fbfr, int num_payinfos)
444 {
445    char temp_string[BUFSIZ];
446    int i;
447
448    add_fld(fbfr, "PAYINFO_INFO_OCC", (char *) &num_payinfos,
(FLDLEN32) sizeof(num_payinfos));
449
450    for (i=0; i < num_payinfos; i++)
451    {
452       sprintf(temp_string, sample_payinfo.name);
453       add_fld(fbfr, "NAME", temp_string, (FLDLEN32)
strlen(temp_string));
454
455       sprintf(temp_string, sample_payinfo.obj_type);
456       add_fld(fbfr, "OBJ_TYPE", temp_string, (FLDLEN32)
strlen(temp_string));
457
458  //     add_invinfo(fbfr, NUM_INVINFO);
459     add_ccinfo(fbfr, NUM_CCINFO);
460
461    }
462 }
463 void add_invinfo(FBFR32 *fbfr, int num_invinfos)
464 {
465    char temp_string[BUFSIZ];
466    int i;
467
468   add_fld(fbfr, "INV_INFO_OCC", (char *) &num_invinfos, (FLDLEN32)
```

```
sizeof(num_invinfos));
469
470    for (i=0; i < num_invinfos; i++)
471    {
472       sprintf(temp_string, sample_invinfo.name);
473       add_fld(fbfr, "NAME", temp_string, (FLDLEN32)
strlen(temp_string));
474
475       sprintf(temp_string, sample_invinfo.address);
476       add_fld(fbfr, "ADDRESS", temp_string, (FLDLEN32)
strlen(temp_string));
477
478       sprintf(temp_string, sample_invinfo.city);
479       add_fld(fbfr, "CITY", temp_string, (FLDLEN32)
strlen(temp_string));
480
481       sprintf(temp_string, sample_invinfo.state);
482       add_fld(fbfr, "STATE", temp_string, (FLDLEN32)
strlen(temp_string));
483
484       sprintf(temp_string, sample_invinfo.zip);
485       add_fld(fbfr, "ZIP", temp_string, (FLDLEN32)
strlen(temp_string));
486
487      sprintf(temp_string, sample_invinfo.country);
488       add_fld(fbfr, "COUNTRY", temp_string, (FLDLEN32)
strlen(temp_string));
489
490       sprintf(temp_string, sample_invinfo.email_addr);
491       add_fld(fbfr, "EMAIL_ADDR", temp_string, (FLDLEN32)
strlen(temp_string));
492
493       add_fld(fbfr, "DELIVERY_PREFER",
&sample_invinfo.delivery_prefer, (FLDLEN32)
sizeof(sample_invinfo.delivery_prefer));
494
495       sprintf(temp_string, sample_invinfo.delivery_descr);
496       add_fld(fbfr, "DELIVERY_DESCR", temp_string, (FLDLEN32)
strlen(temp_string));
497
498       add_fld(fbfr, "INV_TERMS", &sample_invinfo.inv_terms,
(FLDLEN32) sizeof(sample_invinfo.inv_terms));
499
500       sprintf(temp_string, sample_invinfo.inv_instr);
501       add_fld(fbfr, "INV_INSTR", temp_string, (FLDLEN32)
strlen(temp_string));
502
503      add_fld(fbfr, "PO_TYPE", &sample_invinfo.po_type, (FLDLEN32)
sizeof(sample_invinfo.po_type));
504
505       add_fld(fbfr, "PO_OPTIONS", &sample_invinfo.po_options,
(FLDLEN32) sizeof(sample_invinfo.po_options));
506
```

```
507       add_fld(fbfr, "PO_AMOUNT", &sample_invinfo.po_amount,
(FLDLEN32) sizeof(sample_invinfo.po_amount));
508
509       add_fld(fbfr, "PO_BAL_THRESHOLD",
&sample_invinfo.po_bal_threshold, (FLDLEN32)
sizeof(sample_invinfo.po_bal_threshold));
510
511
512    }
513 }
514
515 void add_ccinfo(FBFR32 *fbfr, int num_ccinfos)
516 {
517    char temp_string[BUFSIZ];
518    int i;
519
520   add_fld(fbfr, "CC_INFO_OCC", (char *) &num_ccinfos, (FLDLEN32)
sizeof(num_ccinfos));
521
522    for (i=0; i < num_ccinfos; i++)
523    {
524       sprintf(temp_string, sample_ccinfo.name);
525       add_fld(fbfr, "NAME", temp_string, (FLDLEN32)
strlen(temp_string));
526
527       sprintf(temp_string, sample_ccinfo.address);
528       add_fld(fbfr, "ADDRESS", temp_string, (FLDLEN32)
strlen(temp_string));
529
530       sprintf(temp_string, sample_ccinfo.city);
531       add_fld(fbfr, "CITY", temp_string, (FLDLEN32)
strlen(temp_string));
532
533       sprintf(temp_string, sample_ccinfo.state);
534       add_fld(fbfr, "STATE", temp_string, (FLDLEN32)
strlen(temp_string));
535
536       sprintf(temp_string, sample_ccinfo.zip);
537       add_fld(fbfr, "ZIP", temp_string, (FLDLEN32)
strlen(temp_string));
538
539      sprintf(temp_string, sample_ccinfo.country);
540       add_fld(fbfr, "COUNTRY", temp_string, (FLDLEN32)
strlen(temp_string));
541
542       sprintf(temp_string, sample_ccinfo.debit_num);
543       add_fld(fbfr, "DEBIT_NUM", temp_string, (FLDLEN32)
strlen(temp_string));
544
545       sprintf(temp_string, sample_ccinfo.debit_exp);
546       add_fld(fbfr, "DEBIT_EXP", temp_string, (FLDLEN32)
strlen(temp_string));
547    }
```

```
548 }
549
550 void add_nameinfo(FBFR32 *fbfr, int num_nameinfos)
551 {
552    char temp_string[BUFSIZ];
553    int i;
554
555    add_fld(fbfr, "NAMEINFO_INFO_OCC", (char *) &num_nameinfos,
(FLDLEN32) sizeof(num_nameinfos));
556
557    for (i=0; i < num_nameinfos; i++)
558    {
559       sprintf(temp_string, sample_nameinfo.lastname);
560       add_fld(fbfr, "LAST_NAME", temp_string, (FLDLEN32)
strlen(temp_string));
561
562       sprintf(temp_string, sample_nameinfo.address);
563       add_fld(fbfr, "ADDRESS", temp_string, (FLDLEN32)
strlen(temp_string));
564
565       sprintf(temp_string, sample_nameinfo.city);
566       add_fld(fbfr, "CITY", temp_string, (FLDLEN32)
strlen(temp_string));
567
568       sprintf(temp_string, sample_nameinfo.state);
569       add_fld(fbfr, "STATE", temp_string, (FLDLEN32)
strlen(temp_string));
570
571       sprintf(temp_string, sample_nameinfo.zip);
572       add_fld(fbfr, "ZIP", temp_string, (FLDLEN32)
strlen(temp_string));
573
574       sprintf(temp_string, sample_nameinfo.email_addr);
575       add_fld(fbfr, "EMAIL_ADDR", temp_string, (FLDLEN32)
strlen(temp_string));
576
577
578
579       sprintf(temp_string, sample_nameinfo.salutation);
580       add_fld(fbfr, "SALUTATION", temp_string, (FLDLEN32)
strlen(temp_string));
581
582
583      sprintf(temp_string, sample_nameinfo.title);
584       add_fld(fbfr, "TITLE", temp_string, (FLDLEN32)
strlen(temp_string));
585
586       sprintf(temp_string, sample_nameinfo.contacttype);
587       add_fld(fbfr, "CONTACT_TYPE", temp_string, (FLDLEN32)
strlen(temp_string));
588
589      sprintf(temp_string, sample_nameinfo.company);
590       add_fld(fbfr, "COMPANY", temp_string, (FLDLEN32)
```

```
strlen(temp_string));
591
592       sprintf(temp_string, sample_nameinfo.firstname);
593       add_fld(fbfr, "FIRST_NAME", temp_string, (FLDLEN32)
strlen(temp_string));
594
595       sprintf(temp_string, sample_nameinfo.middlename);
596       add_fld(fbfr, "MIDDLE_NAME", temp_string, (FLDLEN32)
strlen(temp_string));
597
598       sprintf(temp_string, sample_nameinfo.country);
599       add_fld(fbfr, "COUNTRY", temp_string, (FLDLEN32)
strlen(temp_string));
600
601       add_fld(fbfr, "ELEMENT_ID", &sample_nameinfo.element_id,
(FLDLEN32) sizeof(sample_nameinfo.element_id));
602
603    }
604 }
605
606
607
608 void add_phonesinfo(FBFR32 *fbfr, int num_phoneinfos)
609 {
610    char temp_string[BUFSIZ];
611    int i;
612
613    add_fld(fbfr, "PHONES_INFO_OCC", (char *) &num_phoneinfos,
(FLDLEN32) sizeof(num_phoneinfos));
614
615    for (i=0; i < num_phoneinfos; i++)
616    {
617
618      add_fld(fbfr, "TYPE", &sample_phones.type, (FLDLEN32)
sizeof(sample_phones.type));
619
620      sprintf(temp_string, sample_phones.phone);
621       add_fld(fbfr, "PHONE", temp_string, (FLDLEN32)
strlen(temp_string));
622
623
624
625
626    }
627 }
628
629
630 void add_exemptions(FBFR32 *fbfr, int num_exemptions)
631 {
632
633    int i;
634
635   add_fld(fbfr, "EXEMPTIONS_INFO_OCC", (char *) &num_exemptions,
```

```
(FLDLEN32) sizeof(num_exemptions));
636
637    for (i=0; i < num_exemptions; i++)
638    {
639
640      add_fld(fbfr, "TYPE", &sample_exemptions.type, (FLDLEN32)
sizeof(sample_exemptions.type));
641
642      add_fld(fbfr, "PERCENT", &sample_exemptions.percent,
(FLDLEN32) sizeof(sample_exemptions.percent));
643
644
645
646    }
647 }
648
649
650 void add_actginfo(FBFR32 *fbfr, int num_actginfo)
651 {
652
653    int i;
654
655    add_fld(fbfr, "ACTGINFO_INFO_OCC", (char *) &num_actginfo,
(FLDLEN32) sizeof(num_actginfo));
656
657    for (i=0; i < num_actginfo; i++)
658    {
659
660
661       add_fld(fbfr, "ACTG_FUTURE_DOM",
&sample_actginfo.actg_future_dom, (FLDLEN32)
sizeof(sample_actginfo.actg_future_dom));
662
663       add_fld(fbfr, "ACTG_TYPE", &sample_actginfo.actg_type,
(FLDLEN32) sizeof(sample_actginfo.actg_type));
664    }
665 }
666
667
668 void add_locales(FBFR32 *fbfr, int num_locales)
669 {
670
671    int i;
672    char temp_string[BUFSIZ];
673
674
675    add_fld(fbfr, "LOCALE_INFO_OCC", (char *) &num_locales,
(FLDLEN32) sizeof(num_locales));
676
677    for (i=0; i < num_locales; i++)
678    {
679
680      sprintf(temp_string, sample_locales.locale);
```

```
681           add_fld(fbfr, "LOCALE", temp_string, (FLDLEN32)
strlen(temp_string));
682
683    }
684 }
```

Step 7.2 Build the Client Program

To build the client program use the eLink Platform utility `buildclient`.

```
buildclient -o create_cust -f create_cust.c
```

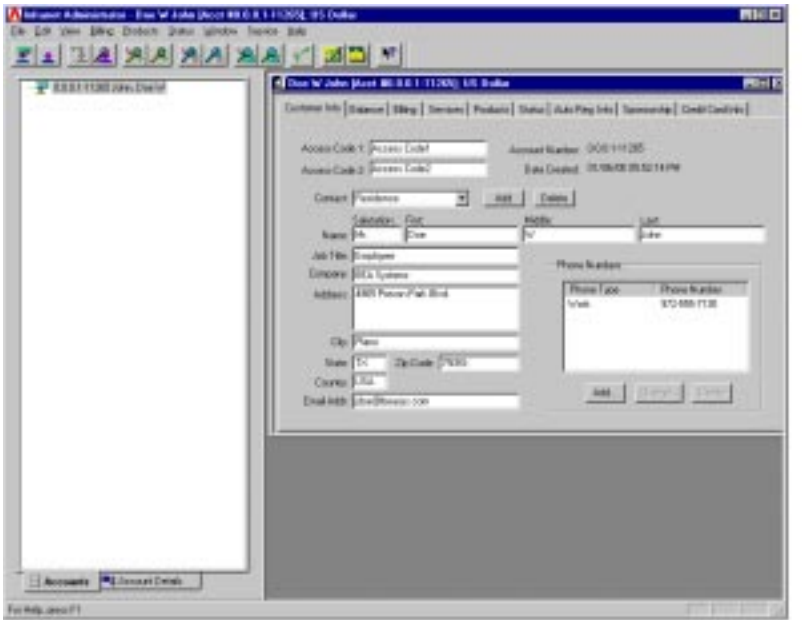
Step 7.3

With eLink Platform booted, the create_cust program can be run by specifying the service to invoke as the first command line parameter. The name of the service (from the adapter configuration file) is CREATE_CUST. Below shows a sample invocation:

```
$ create_cust CREATE_CUST
```

The Portal Infranet Adminstration utility can be used to view the account within the Portal Infranet system. See Figure 5-1.

Figure 5-1 Portal Infranet Administration Utility



Step 8: Shutdown the Application

Run `tmshutdown` to bring the eLink Adapter for Portal Infranet down. Listing 5-11 illustrates this process.

Listing 5-11 Sample Shutdown Procedure

```
$ tmshutdown
Shutdown all admin and server processes? (y/n): y
Shutting down all admin and server process in
/usr/me/simpportal/tuxconfig
Shutting down server processes . . .
  Server Id = 200 Group Id = EPORTALGRP Machine = elink: shutdown
succeeded.
Shutting down admin processes . . .
  Server Id = 0 Group Id = elink Machine = elink: shutdown succeeded.
2 processes stopped.
```

A Portal Infranet Service Interfaces

This section denotes the FML fields expected for each service and the fields that may be returned by the service's response.

The CREATE ACCOUNT Service

In the Configuration File, this service has input fields and the configuration parameters that the service may require. When you create this account, you may be required to set many different fields. There are over 100 fields in the Account object which may be sent in the input buffer. Some of these fields are mandatory, and some are optional. Each of the fields of the account object needs to be in the configuration file.

Table A-1 INPUT Service List

Mandatory/ Optional	Data Type	Input Service Request
M	STR	**NAME**
O	ARRAY	**SERVICES**
M	STR	DEAL_NAME
M	STR	SERVICE_TYPE
M	STR	LOGIN
M	STR	PASSWD_CLEAR

Mandatory/ Optional	Data Type	Input Service Request
O	ARRAY	**BILLINFO**
O	STR(60)	MERCHANT
O	STR(60)	BILL_MODE
O	ENUM	BILL_TYPE
O	UINT	CURRENCY
O	UINT	BILL_WHEN
O	STR(30)	ACCESS_CODE1
O	STR(30)	ACCESS_CODE2
O	ARRAY	**ACTGINFO**
O	UINT	ACTG_FUTURE_DOM
O	UINT	ACTG_TYPE
O	ARRAY	**PAYINFO**
O	STR	NAME
M	STR	OBJ_TYPE
M	ARRAY	**INV_INFO**
M	STR(60)	NAME
M	STR(255)	ADDRESS
M	STR(30)	CITY
M	STR(30)	STATE
M	STR(12)	ZIP
M	STR(60)	COUNTRY
O	STR	EMAIL_ADDR
M	ENUM	DELIVERY_PREFER
M	STR(255)	DELIVERY_DESCR
O	ENUM	INV_TERMS
O	STR(1023)	INV_INSTR
O	ENUM	PO_TYPE
O	ENUM	PO_OPTIONS
O	NUM	PO_AMOUNT
O	NUM	PO_BAL_THRESHOLD
O	TSTAMP	PO_EXP
O	TSTAMP	PO_EXP_THRESHOLD
O	STR(255)	PO_ORDER_NO
O	STR(255)	PO_TERMS
M	ARRAY	**CC_INFO**
M	STR(60)	NAME
M	STR(255)	ADDRESS
M	STR(30)	CITY
M	STR(30)	STATE
M	STR(12)	ZIP
M	STR(60)	COUNTRY
M	STR(30)	DEBIT_NUM
M	STR(30)	DEBIT_EXP

Mandatory/ Optional	Data Type	Input Service Request
O	ARRAY	**NAMEINFO**
M	UINT	ELEMENTID
O	STR(255)	CONTACT_TYPE
O	STR(25)	SALUTATION
O	STR(60)	LAST_NAME
O	STR(30)	FIRST_NAME
O	STR(30)	MIDDLE_NAME
O	STR(60)	TITLE
O	STR(60)	COMPANY
O	STR(255)	ADDRESS
O	STR(30)	CITY
O	STR(30)	STATE
O	STR(12)	ZIP
O	STR(60)	COUNTRY
O	STR(1023)	EMAIL_ADDR
O	ARRAY	**PHONES**
M	ENUM	TYPE
M	STR(25)	PHONE
M	ARRAY	**EXEMPTIONS**
O	ENUM	TYPE
O	NUM	PERCENT
O	ARRAY	**LOCALES**
M	STR(255)	LOCALE

Table A-2 OUTPUT Service Request

Mandatory/ Optional	Data Type	Output Service Response
M	UINT	**OPS_ERROR**
O	STR	**DESCR**
M	STR	**ACCOUNT_NO**
O	ARRAY	**SERVICES**
M	STR	LOGIN
O	STR	PASSWD_CLEAR

Mandatory/ Optional	Data Type	Output Service Response
O	ARRAY	**BILLINFO**
O	STR(60)	MERCHANT
O	STR(60)	BILL_MODE
O	ENUM	BILL_TYPE
O	UINT	CURRENCY
O	UINT	BILL_WHEN
O	STR(30)	ACCESS_CODE1
O	STR(30)	ACCESS_CODE2
O	TSTAMP	LAST_BILL_T
O	TSTAMP	NEXT_BILL_T
O	ARRAY	**ACTGINFO**
O	UINT	ACTG_FUTURE_DOM
O	UINT	ACTG_TYPE
O	ARRAY	**PAYINFO**
O	STR	NAME
M	STR	OBJ_TYPE
M	ARRAY	**INV_INFO**
M	STR(60)	NAME
M	STR(255)	ADDRESS
M	STR(30)	CITY
M	STR(30)	STATE
M	STR(12)	ZIP
M	STR(60)	COUNTRY
O	STR	EMIAL_ADDR
M	ENUM	DELIVERY_PREFER
M	STR(255)	DELIVERY_DESCR
O	ENUM	INV_TERMS
O	STR(1023)	INV_INSTR
O	ENUM	PO_TYPE
O	ENUM	PO_OPTIONS
O	NUM	PO_AMOUNT

Mandatory/ Optional	Data Type	Output Service Response
O	NUM	PO_BAL_THRESHOLD
O	TSTAMP	PO_EXP
O	TSTAMP	PO_EXP_THRESHOLD
O	STR(255)	PO_ORDER_NO
O	STR(255)	PO_TERMS
M	ARRAY	**CCINFO**
M	STR(60)	NAME
M	STR(255)	ADDRESS
M	STR(30)	CITY
M	STR(30)	STATE
M	STR(12)	ZIP
M	STR(60)	COUNTRY
M	STR(30)	DEBIT_NUM
M	STR(30)	DEBIT_EXP
O	ARRAY	**NAMEINFO**
O	STR(255)	CONTACT_TYPE
O	STR(25)	SALUTATION
O	STR(60)	LAST_NAME
O	STR(30)	FIRST_NAME
O	STR(30)	MIDDLE_NAME
O	STR(60)	TITLE
O	STR(60)	COMPANY
O	STR(255)	ADDRESS
O	STR(30)	CITY
O	STR(30)	STATE
O	STR(12)	ZIP
O	STR(60)	COUNTRY
O	STR(1023)	EMAIL_ADDR
O	ARRAY	**PHONES**
O	ENUM	TYPE
O	STR(25)	PHONES
O	ARRAY	**EXAMPTIONS**
M	ENUM	TYPE
O	NUM	PERCENT
O	ARRAY	**LOCALES**
M	STR(255)	LOCALE

The DEL_PROD Service

The DELETE service deletes a business object from the Portal Infranet database. The information required for this operation is fetched from the configuration file. This process cancels the product indicated in that account. Use the tables A3 and A4 to help you delete products in an account.

Table A-3 INPUT Service List

Mandatory/ Optional	Data Type	Input Service Request
M	STR	**ACCOUNT_NO**
M	STR	**PROD_NAME**
O	NUM	**QUANTITY**

Table A-4 OUTPUT Service List

Mandatory/ Optional	Data Type	Output Service Response
M	UINT	**OPS_ERROR**
O	STR	**DESCRO**

The DEL_CUST Service

The DEL_CUST (Delete Customer) service deletes a customer's account from Portal Infranet. The customer's account number (ACCOUNT_NO) must be specified as input. OPS_ERROR and DESCR fields indicate the success or failure of the DEL_CUST service. Use Tables A5 and A6 to help you delete a customer's account.

Table A-5 INPUT Service List

Mandatory/ Optional	Data Type	Input Service Request
M	STR	**SERVICE_TYPE**
M	STR	**LOGIN**
M	STR	**PASSWD_CLEAR**

The MODIFY PAYMENT Service

The MODIFY service options modify the business object in the Portal Infranet database. These services include Account Modification services, and they modify an account information entry in the database and perform modifications in account information involving credit, billing, contact, payment, etc. In the configuration file, this service includes input fields and the configuration parameters that the service may require. The Account Modification services include:

- Modify Payment Information Service
- Modify Billing Information Service
- Modify Credit Balance Element
- Modify Accounting Information Servicen
- Modify Locale Information Service
- Modify Contact Information Service
- Adding New Contacts Service
- Modify Account Status Service

Table A-6 through Table A-20 show the various fields which arrive as input and are listed here per specific service requirement.

Modify Payment Information Service

Table A-6 INPUT Service List

Mandatory/ Optional	Data Type	Input Service Request
M	STR	**ACCOUNT_NO**
M	ARRAY	**PAYINFO**
O	STR	NAME
M	STR	OBJ_TYPE
M	ARRAY	**INV_INFO**
M	STR (60)	NAME
M	STR(255)	ADDRESS
M	STR(30)	CITY
M	STR(30)	STATE
M	STR(12)	ZIP
M	STR(60)	COUNTRY
O	STR	EMAIL_ADDR
M	ENUM	DELIVERY_PREFER
M	STR(255)	DELIVERY_DESCR
O	ENUM	INV_TERMS
O	STR(1023)	INV_INSTR
O	ENUM	PO_TYPE
O	ENUM	PO_OPTIONS
O	NUM	PO_AMOUNT
O	NUM	PO_BAL_THRESHOLD
O	TSTAMP	PO_EXP
O	TSTAMP	PO_EXP_THRESHOLD
O	STR(255)	PO_ORDER_NO
O	STR(255)	PO_TERMS
O	ARRAY	**CC_INFO**
M	STR(60)	NAME
M	STR(255)	ADDRESS
M	STR(30)	CITY
M	STR(30)	STATE
M	STR(12)	ZIP
M	STR(60)	COUNTRY
M	STR(30)	DEBIT_NUM
M	STR(30)	DEBIT_EXP

Modify Billing Information Service

Table A-7 INPUT Service List

Mandatory/ Optional	Data Type	Input Service Request
M	STR	**ACCOUNT_NO**
M	ARRAY	**BILLINFO**
O	ENUM	BILL_TYPE
O	UINT	CURRENCY
O	UINT	BILL_WHEN
O	STR(30)	ACCESS_CODE1
O	STR(30)	ACCESS_CODE2

Table A-8 OUTPUT Service List

Mandatory/ Optional	Data Type	Output Service Response
M	UINT	**OPS_ERROR**
O	STR	**DESCR**

Modify Credit Balance Element

Table A-9 INPUT Service List

Mandatory/ Optional	Data Type	Input Service Request
M	STR	**ACCOUNT_NO**
M	ARRAY	**LIMIT**
M	NUM	CREDIT_LIMIT
O	NUM	CREDIT_FLOOR
O	UINT	CREDIT_THRESHOLDS

Table A-10 OUTPUT Service List

Mandatory/ Optional	Data Type	Output Service Response
M	UINT	**OPS_ERROR**
O	STR	**DESCR**

Modify Accounting Information Service

Table A-11 INPUT Service List

Mandatory/ Optional	Data Type	Input Service Request
M	STR	**ACCOUNT_NO**
O	ARRAY	**ACTGINFO**
M	UINT	ACTG_FUTURE_DOM
O	UINT	ACTG_TYPE

Table A-12 OUTPUT Service List

Mandatory/ Optional	Data Type	Output Service Response
M	UINT	**OPS_ERROR**
O	STR	**DESCR**

Modify Locale Information Service

Table A-13 INPUT Service List

Mandatory/ Optional	Data Type	Input Service Request
M	STR	**ACCOUNT_NO**
O	ARRAY	**LOCALES**
M	STR(255)	LOCALE

Table A-14 OUTPUT Service List

Mandatory/ Optional	Data Type	Output Service Response
M	UINT	**OPS_ERROR**
O	STR	**DESCR**

Modify Contact Information Service

Table A-15 INPUT Service List

Mandatory/ Optional	Data Type	Input Service Request
M	STR	**ACCOUNT_NO**

Mandatory/ Optional	Data Type	Input Service Request
O	ARRAY	**NAMEINFO**
M	UINT	ELEMENT_ID
O	STR(255)	CONTACT TYPE
O	STR(25)	SALUTATION
O	STR(60)	LAST NAME
O	STR(30)	FIRST NAME
O	STR(30)	MIDDLE NAME
O	STR(60)	TITLE
O	STR(60)	COMPANY
O	STR(255)	ADDRESS
O	STR(30)	CITY
O	STR(30)	STATE
O	STR(12)	ZIP
O	STR(60)	COUNTRY
O	STR(1023)	EMAIL_ADDR
O	ARRAY	PHONES
M	ENUM	TYPE
O	STR	PHONE

Table A-16 OUTPUT Service List

Mandatory/ Optional	Data Type	Output Service Response
M	UINT	OPS_ERROR
O	STR	DESCR

Adding New Contacts Service

Table A-17 INPUT Service List

Mandatory/ Optional	Data Type	Input Service Request
M	STR	**ACCOUNT_NO**
O	ARRAY	**NAMEINFO**
M	UINT	ELEMENT_ID
O	STR(255)	CONTACT TYPE
O	STR(25)	SALUTATION
O	STR(60)	LAST NAME
O	STR(30)	FIRST NAME
O	STR(30)	MIDDLE NAME
O	STR(60)	TITLE
O	STR(60)	COMPANY
O	STR(255)	ADDRESS
O	STR(30)	CITY
O	STR(30)	STATE
O	STR(12)	ZIP
O	STR(60)	COUNTRY
O	STR(1023)	EMAIL_ADDR
O	ARRAY	PHONES
M	ENUM	TYPE
O	STR	PHONE

Table A-18 OUTPUT Service List

Mandatory/ Optional	Data Type	Output Service Response
M	UINT	**OPS_ERROR**
O	STR	**DESCR**

Modify Account Status Service

Table A-19 INPUT Service List

Mandatory/ Optional	Data Type	Input Service Request
M	STR	**ACCOUNT_NO**
O	STR	**FLD_DESCR**
M	ENUM	**STATUS**
M	UINT	**STATUS_FLAGS**
	TSTAMP	**CLOSE_WHEN_T**

Table A-20 OUTPUT Service List

Mandatory/ Optional	Data Type	Output Service Response
M	UINT	**OPS_ERROR**
O	STR	**DESCR**

The ORDER HANDLING Service

These services come under the category of Order Handling services, and they modify account information for an order-related entry in the database. With these services you perform modifications in account information in Portal Infranet about Deal, Plan, Service, and other services to the account.

In the configuration file, this service has input fields and configuration parameters that the service may require. The various fields required to modify different aspects of the order varies according to the service. At the time of account modification, you are required to set the fields in the input buffer. It is mandatory to set some of these fields, while others are optional. Each of the object order fields can be sent as part of the input buffer and should be included in the configuration file. The Order Handling services include:

- Add Deal to Account Service
- Modify Products In Account Service
- Add Service to Account Service
- Modify Product Status Service

Table A-6 through Table A-20 show the various fields which arrive as input and are listed here per specific service requirement.

Add Deal to Account Service

Table A-21 INPUT Service List

Mandatory/ Optional	Data Type	Input Service Request
M	STR	**ACCOUNT_NO**
M	STR	**DEAL_NAME**

Table A-22 OUTPUT Service List

Mandatory/ Optional	Data Type	Output Service Response
M	UINT	**OPS_ERROR**
O	STR	**DESCR**

Modify Products In Account Service

Table A-23 INPUT Service List

Mandatory/ Optional	Data Type	Input Service Request
M	STR	**ACCOUNT_NO**
O	NUM	**PURCHASE_DISCOUNT**
O	NUM	**USAGE_DISCOUNT**
O	TSTAMP	**PURCHASE_START_T**
O	TSTAMP	**PURCHASE_END_T**
O	TSTAMP	**USAGE_START_T**
O	TSTAMP	**USAGE_END_T**
M	STR	**PROD_NAME**

Table A-24 OUTPUT Service List

Mandatory/ Optional	Data Type	Output Service Response
M	UINT	**OPS_ERROR**
O	STR	**DESCR**

Add Service to Account Service

Table A-25 INPUT Service List

Mandatory/ Optional	Data Type	Input Service Request
M	STR	**ACCOUNT_NO**
M	STR	**PLAN_NAME**
M	ARRAY	**SERVICES**
M	STR	DEAL_NAME
M	STR	SERVICE_TYPE
M	STR	FLD_LOGIN
M	STR	PASSWD_CLEAR

Table A-26 OUTPUT Service List

Mandatory/ Optional	Data Type	Output Service Response
M	UINT	**OPS_ERROR**
O	STR	**DESCR**

Modify Product Status Service

Table A-27 INPUT Service List

Mandatory/ Optional	Data Type	Input Service Request
M	STR	**ACCOUNT_NO**
M	ENUM	**STATUS**
M	STR	**STATUS_FLAGS**
M	STR	**PROD_NAME**

Table A-28 OUTPUT Service List

Mandatory/ Optional	Data Type	Output Service Response
M	UINT	**OPS_ERROR**
O	STR	**DESCR**

The Modification Service Handling Service

The Modification Service Handling services are account maintenance services that modify account information for service related entry in the database. These services perform modifications in account information in Portal Infranet about user password, login ID, and status of the services. Modification Service Handling functionality includes user login, password, and service status. The Modification Service Handling services include:

- Modify (Service) Login Information Service
- Modify (Service) Password Service
- Modify (Service) Status Service

Modify (Service) Login Information Service

Table A-29 INPUT Service List

Mandatory/ Optional	Data Type	Input Service Request
M	STR	**ACCOUNT_NO**
O	STR	**SERVICE_TYPE**
M	STR	**LOGIN**
M	STR	**NAME**

Table A-30 OUTPUT Service List

Mandatory/ Optional	Data Type	Output Service Response
M	UINT	**OPS_ERROR**
O	STR	**DESCR**

Modify (Service) Password Service

Table A-31 INPUT Service List

Mandatory/ Optional	Data Type	Input Service Request
M	STR	**ACCOUNT_NO**
O	STR	**SERVICE_TYPE**
M	STR	**LOGIN**
M	STR	**PASSWD_CLEAR**

Table A-32 OUTPUT Service List

Mandatory/ Optional	Data Type	Output Service Response
M	UINT	**OPS_ERROR**
O	STR	**DESCR**

Modify (Service) Status Service

Table A-33 INPUT Service List

Mandatory/ Optional	Data Type	Input Service Request
M	STR	**ACCOUNT_NO**
M	STR	**SERVICE_TYPE**
M	STR	**LOGIN**
	STR	**DESCR**

Mandatory/ Optional	Data Type	Input Service Request
M	ENUM	**STATUS**
M	UINT	**STATUS_FLAGS**
O	TSTAMP	**CLOSE_WHEN_T**

Table A-34 OUTPUT Service List

Mandatory/ Optional	Data Type	Output Service Response
M	UINT	**OPS_ERROR**
O	STR	**DESCR**

The Get Service

The Get services are billing services that get billing information from the database in order to perform modifications in account information in Portal Infranet for *get bills*, line items, and invoice image service instances to the account. In the configuration file, this service includes input fields and configuration parameters that the service requires.

The fields that you must modify for various aspects of billing are different for each service. At the time of account modification, you are required to set the fields in the input buffer. It is mandatory to set some of these fields, while others are optional. Each of the object order fields can be sent as part of the input buffer and should be included in the configuration file. The Get services include:

- Get List of Bills for an Account Service
- Get the Line Items for a Bill Service
- Get Invoice Image for a Bill Service
- Get Account No Service

Get List of Bills for an Account Service

Table A-35 INPUT Service List

Mandatory/ Optional	Data Type	Input Service Request
M	STR	**ACCOUNT_NO**

Table A-36 OUTPUT Service List

Mandatory/ Optional	Data Type	Output Service Response
M	UINT	**OPS_ERROR**
O	STR	**DESCR**

Mandatory/ Optional	Data Type	Output Service Response
M	STR	**ACCOUNT_NO**
O	ARRAY	**BILLS**
O	STR	BILL_NO
O	TSTAMP	START_T
O	TSTAMP	END_T
O	NUM	TOTAL_DUE
O	NUM	CURRENT_TOTAL
O	UINT	CURRENCY

Get the Line Items for a Bill Service

Table A-37 INPUT Service List

Mandatory/ Optional	Data Type	Input Service Request
M	STR	**ACCOUNT_NO**
M	ARRAY	**BILLS**
M	STR	BILL_NO
O	TSTAMP	START_T
O	TSTAMP	END_T
O	ENUM	STATUS

Table A-38 OUTPUT Service List

Mandatory/ Optional	Data Type	Output Service Response
M	UINT	**OPS_ERROR**
O	STR	**DESCR**
M	STR	**ACCOUNT_NO**

Mandatory/ Optional	Data Type	Output Service Response
O	ARRAY	**BILLS**
O	STR	BILL_NO
O	TSTAMP	START_T
O	TSTAMP	END_T
O	NUM	TOTAL_DUE
O	NUM	CURRENT_TOTAL
O	UINT	CURRENCY
O	ARRAY	ITEMS
O	STR	ITEM_NO
O	STR	NAME
O	NUM	ITEM_TOTAL
O	UINT	CURRENCY

Get Invoice Image for a Bill Service

Table A-39 INPUT Service List

Mandatory/ Optional	Data Type	Input Service Request
M	STR	**ACCOUNT_NO**
M	ARRAY	**BILLS**
M	STR	BILL_NO

Table A-40 OUTPUT Service List

Mandatory /Optional	Data Type	Output Service Response
M	UINT	**OPS_ERROR**
O	STR	**DESCR**
M	STR	**ACCOUNT_NO**
O	ARRAY	**BILLS**
O	STR	BILL_NO
O	BUF	BUFFER

GET_ACCOUNT_NO Service

Table A-41 INPUT Service List

Mandatory/ Optional	Data Type	Input Service Request
M	STR	**SERVICE_TYPE**
M	STR	**LOGIN**

Table A-42 OUTPUT Service List

Mandatory/ Optional	Data Type	Output Service Response
M	STR	**ACCOUNT_NO**
M	STR	**OPS_ERROR**
O	STR	**DESCR**

The Querying Services

The Querying services query account information in Portal Infranet about the account and product, and they rate the related entry in the database. A query might be a single operation from the Portal Infranet point of view or a collection of read operations.

The fields that you must modify for various aspects of querying are different for each service. At the time of account querying, you are required to set the fields in the input buffer. It is mandatory to set some of these fields, while others are optional. Each of the object order fields can be sent as part of the input buffer and should be included in the configuration file. The Query services include:

- Get Account Information Service
- Get Product Information Service
- Get Rate Information Service

Get Account Information Service

Table A-43 INPUT Service List

Mandatory/ Optional	Data Type	Input Service Request
M	STR	**ACCOUNT_NO**

Table A-44 OUTPUT Service List

Mandatory/ Optional	Data Type	Output Service Response
M	UINT	**OPS_ERROR**
O	STR	**DESCR**
M	STR	**ACCOUNT_NO**
O	ARRAY	**SERVICES**
M	STR	LASTSTAT_CMNT
M	TSTAMP	LAST_STATUS_T
O	UINT	STATUS_FLAGS
O	ENUM	STATUS
M	STR	LOGIN
O	STR	PASSWORD
O	ARRAY	**BILLINFO**
O	STR(60)	MERCHANT
O	STR(60)	BILL_MODE
O	ENUM	BILL_TYPE
O	UINT	CURRENCY
O	UINT	BILL_WHEN
O	STR(30)	ACCESS_CODE1
O	STR(30)	ACCESS_CODE2
O	TSTAMP	LAST_BILL_T
O	TSTAMP	NEXT_BILL_T
O	ARRAY	**ACTGINFO**
M	UINT	ACTG_FUTURE_DOM
O	UINT	ACTG_TYPE

Mandatory/ Optional	Data Type	Output Service Response
O	ARRAY	**PAYINFO**
O	STR	NAME
M	STR	OBJ_TYPE
M	ARRAY	**INV_INFO**
M	STR(60)	NAME
M	STR(255)	ADDRESS
M	STR(30)	CITY
M	STR(30)	STATE
M	STR(12)	ZIP
M	STR(60)	COUNTRY
O	STR	EMAIL_ADDR
M	ENUM	DELIVERY_PREFER
M	STR(255)	DELIVERY_DESCR
O	ENUM	INV_TERMS
O	STR(1023)	INV_INSTR
O	ENUM	PO_TYPE
O	ENUM	PO_OPTIONS
O	NUM	PO_AMOUNT
O	NUM	PO_BAL_THRESHOLD
O	TSTAMP	PO_EXP
O	TSTAMP	PO_EXP_THRHESHOLD
O	STR(255)	PO_ORDER_NO
O	STR(255)	PO_TERMS
M	ARRAY	**CC_INFO**
M	STR(60)	NAME
M	STR(255)	ADDRESS
M	STR(30)	CITY
M	STR(30)	STATE
M	STR(12)	ZIP
M	STR(60)	COUNTRY
M	STR(30)	DEBIT_NUM
M	STR(30)	DEBIT_EXP

Mandatory/ Optional	Data Type	Output Service Response
O	ARRAY	**NAMEINFO**
O	STR(255)	CONTACT_TYPE
O	STR(25)	SALUTATION
O	STR(60)	LAST_NAME
O	STR(30)	FIRST_NAME
O	STR(30)	MIDDLE_NAME
O	STR(60)	TITLE
O	STR(60)	COMPANY
O	STR(255)	ADDRESS
O	STR(30)	CITY
O	STR(30)	STATE
O	STR(12)	ZIP
O	STR(60)	COUNTRY
O	STR(1023)	EMAIL_ADDR
O	ARRAY	**PHONES**
M	ENUM	TYPE
M	STR(25)	PHONE
O	ARRAY	**EXAMPTIONS**
M	ENUM	TYPE
O	NUM	PERCENT
O	ARRAY	**LOCALES**
M	STR(255)	LOCALE
O	ARRAY	**BALANCES**
O	NUM	CREDIT_FLOOR
O	NUM	CREDIT_LIMIT
O	UINT	CREDIT_THRESHOLDS
O	NUM	CREDIT_CURRENT_BAL
O	UINT	FLAGS
O	NUM	OPEN_BAL

Mandatory/ Optional	Data Type	Output Service Response
O	ARRAY	**PRODUCTS**
O	TSTAMP	CREATED_T
O	NUM	CYCLE_DISCOUNT
O	NUM	CYCLE_DISC_AMT
O	TSTAMP	CYCLE_END_T
O	TSTAMP	CYCLE_START_T
O	STR	DEAL_NAME
O	UINT	FLAGS
O	TSTAMP	MMC_END_T
O	TSTAMP	MMC_START_T
O	UINT	MMC_TYPE
O	NUM	PURCHASE_DISCOUNT
O	NUM	PURCHASE_DISC_AMT
O	TSTAMP	PURCHASE_END_T
O	TSTAMP	PURCHASE_START_T
O	STR	SERVICE_TYPE
O	STR	LOGIN
O	STR	PASSWORD_CLEAR

Get Product Information Service

Table A-45 INPUT Service List

Mandatory/ Optional	Data Type	Input Service Request
M	STR	**ACCOUNT_NO**

Table A-46 OUTPUT Service List

Mandatory/ Optional	Data Type	Output Service Response
M	UINT	**OPS_ERROR**
O	STR	**DESCR**
O	ARRAY	**PRODUCTS**
O	STR	FIELD_NAME
O	STR	FIELD_DESCR

Get Rate Information Service

Table A-47 INPUT Service List

Mandatory/ Optional	Data Type	Input Service Request
M	STR	**ACCOUNT_NO**
M	STR	**PROD_NAME**

Table A-48 OUTPUT Service List

Mandatory/ Optional	Data Type	Output Service Response
M	UINT	**OPS_ERROR**
O	UINT	**CURRENCY**
O	STR	**DESCR**
O	TSTAMP	**END_T**
O	STR	**NAME**
O	UINT	**PRIORITY**
O	NUM	**PRIORITY_FIRST**
O	NUM	**PRIORITY_LAST**
O	UINT	**PURCHASE_MAX**
O	UINT	**PURCHASE_MIN**
0	TSTAMP	**RELATIVE_END_T**
0	TSTAMP	**RELATIVE_START_T**
0	UINT	**VALID_DOM**

Mandatory/ Optional	Data Type	Output Service Response
0	UINT	**VALID_DOW**
0	UINT	**VALID_MOY**

The GET_AUTH Service

The GET_AUTH (Get Authorization) service requests authorization from Portal Infranet to confirm that a specified login has access to a service type (email, IP, wireless, etc.) that is billed by Portal Infranet. The GET_AUTH service requests Portal Infranet to validate the user's login and password, the user's account status / balance, and whether they are subscribing to the specified service type (email, IP, wireless, etc.) The RESULT field indicates success (value of 1) or failure (value of 0). If the authorization check fails additional information as to why the authorization check failed is returned in the TYPE and DESCR fields.

Table A-49 INPUT Service List

Mandatory/ Optional	Data Type	Input Service Request
M	STR	**SERVICE_TYPE**
M	STR	**LOGIN**
M	STR	**PASSWD_CLEAR**

Table A-50 OUTPUT Service List

Mandatory/ Optional	Data Type	Output Service Response
M	UINT	**RESULT**
O	UINT	**TYPE**
O	STR	**DESCR**

The POST_ACTIVITY Service

The POST_ACTIVITY Service is used to record usage activity of a service type that is billed by Portal Infranet. For example, if Portal Infranet is used to bill wireless telephone customers, then usage of a wireless phone can be posted to Portal Infranet using the POST_ACTIVITY service. The SERVICE_TYPE and LOGIN fields specify that a customer (LOGIN) was using a service type (IP, wireless, email etc.)

The NAME field indicates the name of the rate to apply to the account for this usage. For example "ip/dialup/async/hourly" is the name of rate for billing Internet access on an hourly basis. The QUANTITY field is used to indicate how many units of a service type was used (30 hours, 600 minutes, etc). The units (hours, minutes, etc.) of the QUANTITY are implied by the NAME field (the rate to use). OPS_ERROR and DESCR fields are used to indicate the success or failure of the POST_ACTIVITY service.

Table A-51 INPUT Service List

Mandatory/ Optional	Data Type	Input Service Request
M	STR	**SERVICE_TYPE**
M	STR	**LOGIN**
M	STR	**DESCR**
M	STR	**NAME**
M	UINT	**QUANTITY**
O	STR	**IMPACE_CATEGORY**

Table A-52 OUTPUT Service List

Mandatory/ Optional	Data Type	Output Service Response
M	UINT	**OPS_ERROR**
M	STR	**DESCR**

The LOGIN Service

The LOGIN service requests authorization from Portal Infranet to confirm that a specified login has access to a service type (email, IP, wireless, etc.) that is billed by Portal Infranet (see description for the GET_AUTH). If authorization is granted, the LOGIN service starts a session within Portal Infranet to indicate that a customer is using the specified service type.

For example, suppose Portal Infranet is used to bill wireless telephone customers. When a wireless telephone customer dials a number, the LOGIN Service could be used to validate that the customer is valid, that his/her account is paid, etc. If this validation is successful, the LOGIN Service then starts a session within Portal Infranet to reflect the customer's usage of the wireless service. The LOGOUT Service (described below) can later be used to end this session.

The RESULT field indicates success (value of 1) or failure (value of 0). If the authorization check fails additional information as to why the authorization check failed is returned in the TYPE and DESCR fields. If authorization is successful, the field HANDLE contains a string that identifies the session that is started on behalf of the customer. This string needs to be supplied when this session is subsequently terminated with the LOGOUT service.

Table A-53 INPUT Service List

Mandatory/ Optional	Data Type	Input Service Request
M	STR	**SERVICE_TYPE**

Mandatory/ Optional	Data Type	Input Service Request
M	STR	**LOGIN**
M	STR	**PASSWD_CLEAR**
M	STR	**DESCR**
M	TSTAMP	**START_T**

Table A-54 OUTPUT Service List

Mandatory/ Optional	Data Type	Output Service Response
M	UINT	**RESULT**
O	UINT	**TYPE**
O	STR	**DESCR**
M	STR	**HANDLE**

The LOGOUT Service

The LOGOUT service requests that a session in the Portal Infranet system be ended, and that the usage associated with this session be applied to the customer's account. The LOGOUT service is used to end sessions that are started with the LOGIN service.

The HANDLE field should contain a string that specifies the session to end. This string is returned by the LOGIN service or the GET_SESSIONS service (described below). The DESCR field can be used to record a description of this session in Portal Infranet. The END_T (end time) field is used to denote at what time the customer's

session ended. The NAME field should supply a rate name and the QUANTITY field should indicate the number of units this session used. OPS_ERROR and DESCR fields indicate success or failure of the LOGOUT service request.

Table A-55 INPUT Service List

Mandatory/ Optional	Data Type	Input Service Request
M	STR	**HANDLE**
M	STR	**DESCR**
M	STR	**END_T**
M	STR	**NAME**
M	UINT	**QUANTITY**
O	STR	**IMPACE_CATEGORY**

Table A-56 OUTPUT Service List

Mandatory/ Optional	Data Type	Output Service Response
M	UINT	**OPS_ERROR**
O	STR	**DESCR**

The GET_SESSIONS Service

The GET_SESSIONS service returns sessions that are active within Portal Infranet for a specified service type (email, IP, wireless, etc). The SERVICE_TYPE field specifies what type of sessions to return. For every session of that type, the following information is returned: the session ID (HANDLE), start time of the session (START_T), login of session user (LOGIN), service type (SERVICE_TYPE), and account number (ACCOUNT_NO).

Table A-57 INPUT Service List

Mandatory/ Optional	Data Type	Input Service Request
M	STR	**SERVICE_TYPE**

Table A-58 OUTPUT Service List

Mandatory/ Optional	Data Type	Output Service Response
M	STR	**OPS_ERROR**
M	STR	**DESCR**
M	STR	**HANDLE**
M	STR	**START_T**
M	STR	**LOGIN**
M	STR	**SERVICE_TYPE**
M	STR	**ACCOUNT_NO**

B Portal Infranet Enumerations

Table B-1 provides a listing of typical ENUMS used in Portal Infranet services.

Table B-1 List of Enumerations

Enumeration Name	Description
BILL_TYPE	PIN_BILL_TYPE_INVOICE=10001 PIN_BILL_TYPE_CC=10003 PIN_BILL_TYPE_SUBORD=10007
DELIVERY_PREFER	PIN_INV_EMAIL_DELIVERY=0 PIN_INV_USP_DELIVERY=1 PIN_INV_FAX_DELIVERY=2
INV_TERMS	PIN_INV_TERMS_UNDEFINED=0
PO_TYPE	PIN_PO_TYPE_NONE=0 PIN_PO_TYPE_TIME=1 PIN_PO_TYPE_AMOUNT=2 PIN_PO_TYPE_COMBO=3
PO_OPTIONS	PIN_PO_OPTION_CONTINUE=0 PIN_PO_OPTION_INACTIVATE=1

Table B-1 List of Enumerations

Enumeration Name	Description
FLD_TYPE(Phones)	PIN_PHONE_TYPE_UNKNOWN=0 PIN_PHONE_TYPE_HOME=1 PIN_PHONE_TYPE_WORK=2 PIN_PHONE_TYPE_FAX=3 PIN_PHONE_TYPE_PAGER=4 PIN_PHONE_TYPE_PORTABLE=5 PIN_PHONE_TYPE_POP=6 PIN_PHONE_TYPE_SUPPORT=7
FIELD_TYPE(Exceptions)	PIN_RATE_TAX_JUR_FED=0 PIN_RATE_TAX_JUR_STATE=1 PIN_RATE_TAX_JUR_COUNTY=2 PIN_RATE_TAX_JUR_CITY=3 PIN_RATE_TAX_JUR_SEC_COUNTY=4 PIN_RATE_TAX_JUR_SEC_CITY=5 PIN_RATE_TAX_JUR_TERRITORY=6 PIN_RATE_TAX_JUR_SEC_STATE=7 PIN_RATE_TAX_JUR_DISTRICT=8
PIN_FLD_STATUS	PIN_STATUS_DEFUNCT=0 PIN_STATUS_ACTIVE=10100 PIN_STATUS_INACTIVE=10102 PIN_STATUS_CLOSED=10103
BILL_TYPE	PIN_BILL_TYPE_INVOICE=10001 PIN_BILL_TYPE_CC=10003 PIN_BILL_TYPE_SUBORD=10007
DELIVERY_PREFER	PIN_INV_EMAIL_DELIVERY=0 PIN_INV_USP_DELIVERY=1 PIN_INV_FAX_DELIVERY=2
PIN_FLD_TYPE	PIN_BAL_IMPACT_TYPE_INVALID=0 PIN_BAL_IMPACT_TYPE_REBATE=880 PIN_BAL_IMPACT_TYPE_BALANCE=881 PIN_BAL_IMPACT_TYPE_CASH=882
STATUS_FLAGS (This string field consists of enumerations.)	None=0 Waiting for Install = 1 Network Configuration = 2 Both flags = 3

Table B-1 List of Enumerations

Enumeration Name	Description
STATUS_FLAGS	PIN_STATUS_FLAG_MANUAL=04 PIN_STATUS_FLAG_DEBT=02

C Error and Informational Messages

This section contains the following descriptions of error, informational, and warning messages that can be encountered while using the eLink Adapter for Portal Infranet component. ELINK_ADAPTER_ERR_CODE contains a string denoting one of the following error categories:

Error Categories	Description
ELINK_EAPP_API	The application's API returned an error. Note that this refers to the application's API returning an infrastructure level error rather than a business level error.
ELINK_EAPP_UNAVAIL	The application was unavailable.
ELINK_EATMI	An ATMI error occurred.
ELINK_ECONFIG	An error occurred with the adapter configuration data.
ELINK_EFML	An FML error occurred.
ELINK_EINVAL	Invalid value/argument error. For example, an FML32 request buffer is sent to an adapter without all the required FML32 fields being present.
ELINK_EITYPE	An input type mismatch. For example, converting between FML32 and application data types on the input.
ELINK_ELIMIT	Out of range value.

ELINK_ENOENT	No entry found. The application functionality corresponding to the **service** could not be found.
ELINK_EOS	An operating system error. For example, a memory allocation error.
ELINK_EOTYPE	An output type mismatch. For example, converting between **FML32** and application data types on the output.
ELINK_EPERM	A permissions error.
ELINK_EPROTO	A protocol error.
ELINK_ETIME	A timeout error. For example, timing-out while waiting for the application to process the request.
ELINK_ETRAN	A transaction error.

Error Messages

The ELINK_ADAPTER_ERR field contains the complete error message as described in this section. These messages are also logged to the ULOG.

1001 ELINK_EFML" ERROR : %s	**FML32 response buffer could not be reallocated while trying to add %s field for %s group. Exiting from %s.**	
	DESCRIPTION	Adapter was unable to allocate memory for an FML32 response buffer.
	ACTION	Make sure the operating system parameters are set correctly for the amount of memory on the machine and the amount of memory that can be used by a process. Reduce the memory usage on the machine or increase the amount of physical memory on the machine
1002 ELINK_ECONFIG" ERROR : %s	**While opening configuration file %s. Exiting from loadConfigurationInformation.**	
	DESCRIPTION	Adapter was unable to open configuration file specified with -C command line option
	ACTION	Make sure configuration file specified with -C option exists in APPDIR directory. Make sure that the permissions of the file allow the adapter to read the file.
1003 ELINK_ECONFIG" ERROR	**Required parameter -C missing.**	
	DESCRIPTION	The command line option (CLOPT) did not specify the adapter configuration file

	ACTION	Add -C CLOPT parameter for ELINKPORTALO server in UBBCONFIG
1004 ELINK_ECONFIG" ERROR : %s	**Unable to read first %s section. Exiting from loadConfigurationInformation.**	
	DESCRIPTION	The adapter configuration file specified by command line option is missing the specified section.
	ACTION	Correct configuration file according to the format specified in the eLink Adapter for Portal Infranet User's Guide.
1005 ELINK_ECONFIG" ERROR : %s	**Invalid Server name %s . Exiting from loadConfigurationInformation.**	
	DESCRIPTION	Server name specified in the configuration file is not PORTAL_ADAPTER
	ACTION	Modify configuration file to specify adapter name as PORTAL_ADAPTER.
1006 ELINK_ECONFIG" ERROR : %s	**Buffer overflow while reading %s. Exiting from loadConfigurationInformation.**	
	DESCRIPTION	The configuration paramter read from the adapter configuration file was greater than 256 characters.
	ACTION	Correct configuration file.
1007 ELINK_ECONFIG" ERROR : %s	**While trying to close tag handle for configuration file. Exiting from loadConfigurationInformation.**	
	DESCRIPTION	Internal error occurred while reading the adapter configuration file.

	ACTION	Verify that the ADK library (libadk) in use is the original version delivered with the adapter. Contact BEA Customer Support for additional help.
1008 ELINK_ECONFIG" ERROR : %s	**Property given or no value for a property given. Exiting from loadConfigurationInformation.**	
	DESCRIPTION	Specified configuration item was not given a value.
	ACTION	Correct configuration file to specify a value or delete configuration parameter altogether.
1009 ELINK_ECONFIG" ERROR : %s	**Unable to add %s element to %s hash table. Exiting from loadConfigurationInformation.**	
	DESCRIPTION	Adapter was unable to add configuration data to hash table data structure.
	ACTION	Make sure the operating system is configured with enough memory.
1010 ELINK_EATMI" ERROR : %s	**tpadvertise failed for %s service. Exiting from tpsvrinit. ERROR**	
	DESCRIPTION	Adapter was unable to advertise configured service.
	ACTION	Make sure that service name is not duplicated in adapter configuration file. Each service name must be unique up to 15 characters.
1011 ELINK_EAPP_API" ERROR : %s	**Blank %s Flist could not be created for %s service. Exiting from %s.**	
	DESCRIPTION	Adapter was unable to create Portal FLIST data structure.

	ACTION	Ensure that the Portal Infranet version is 5.5.3. Contact BEA Customer Support for additional help.
1012 ELINK_EITYPE" ERROR : %s	**Incoming buffer type is not FML32 for %s service. Exiting from EL_PORTAL_OUT**	
	DESCRIPTION	Request was sent to the adapter that was not of type FML32.
	ACTION	Correct invoking application to allocate and populate an FML32 request message for Portal Infranet services.
1013 ELINK_ENOENT" ERROR : %s	**Service called. Exiting from %s.**	
	DESCRIPTION	Adapter was invoked for a service that it is not configured for.
	ACTION	Verify configuration file contains specified service. Modify invoking application to call correct TUXEDO service.
1014 ELINK_EINVAL" ERROR : %s	**Invalid occurrence field name %s is given in %s. Exiting from %s.**	
	DESCRIPTION	Configuration file specifies an occurrence field that is not defined in an FML32 field table.
	ACTION	Make sure that FML32 field tables are specified in adapter's environment. Make sure that FML field is defined for specified occurrence field.
1015 ELINK_EOTYPE" ERROR : %s	**failed for %s. Exiting from %s.**	
	DESCRIPTION	Failed to convert FLIST data to FML data.

	ACTION	Make sure the type of the FML field is correctly specified in the FML field table. Make sure FIELD_GROUP Definition in configuration file defines specified field.
1016 ELINK_EFML" ERROR : %s	**Unable to delete a %s field in FML in %s. Exiting from %s.**	
	DESCRIPTION	Adapter attempted to call Fdel32() but received an error response.
	ACTION	Make sure specified field is defined in FML field table and in FIELD_GROUP definition.
1017 ELINK_EFML" ERROR : %s	**Unknown field %s to FML buffer(Ferror32 = FBADFLD) in %s. Exiting from %s.**	
	DESCRIPTION	FML32 field is not defined in adapter's environment.
	ACTION	Make sure that FML field tables are specified in adapter's environment file.
1018 ELINK_EAPP_ UNAVAIL" ERROR : %s	**Portal connection could not be made for %s service. Exiting from %s."**	
	DESCRIPTION	Adapter was unable to establish a connection with Portal Infranet.
	ACTION	Verify Portal Infranet is up. Verify pin.conf file exists in adapter's APPDIR. Verify connection parameters specified in pin.conf are correct.
1019 ELINK_EBUSERR" ERROR	**Business logic error for %s service. See FML32 buffer for detail. Exiting from %s.**	
	DESCRIPTION	Error occurred within Portal Infranet while executing specified service.

	ACTION	Verify business logic (account exists for specified request, product exists, etc.) See additional error detail.
1020 ELINK_EAPP_ UNAVAIL" ERROR : %s	**Connection to Portal's Infranet could not be made even after three attempts for %s service. Exiting from %s."**	
	DESCRIPTION	Adapter was unable to establish connection with Portal Infranet.
	ACTION	Verify Portal Infranet is up. Verify pin.conf file exists in adapter's APPDIR. Verify connection parameters specified in pin.conf are correct.
1021 ELINK_ECONFIG" ERROR	**Configuration file could not be loaded.**	
	DESCRIPTION	Error occurred while reading configuration file.
	ACTION	Look at previous error message to determine source of problem.
1022 ELINK_EOS" ERROR	**ELINK_EOS Memory allocation failed for adapter configuration information. Exiting from tpsvrinit.**	
	DESCRIPTION	Adapter was unable to dynamically allocate memory.
	ACTION	Make sure the operating system parameters are set correctly for the amount of memory on the machine and the amount of memory that can be used by a process. Reduce the memory usage on the machine or increase the amount of physical memory on the machine.
1023 ELINK_ECONFIG" ERROR : %s	**Service hash table could not be initialized. Exiting from loadConfigurationInformation.**	

	DESCRIPTION	Adapter attempted to initialize internal data structure but received an error response.
	ACTION	Look for previous errors to discover the cause of the problem.
1024 ELINK_ECONFIG" ERROR : %s	**Group hash table could not be initialized. Exiting from loadConfigurationInformation.**	
	DESCRIPTION	Adapter attempted to initialize internal data structure but received an error response.
	ACTION	Look for previous errors to discover the cause of the problem.
1025 ELINK_ECONFIG" ERROR : %s	**While trying to close tag handle for field groups. Exiting from loadConfigurationInformation.**	
	DESCRIPTION	Internal error occurred while reading the adapter configuration file.
	ACTION	Verify that the ADK library (libadk) in use is the original version delivered with the adapter. Contact BEA Customer Support for additional help.
1026 ELINK_ECONFIG" ERROR : %s	**While trying to close tag handle for services. Exiting from loadConfigurationInformation.**	
	DESCRIPTION	Internal error occurred while reading the adapter configuration file.
	ACTION	Verify that the ADK library (libadk) in use is the original version delivered with the adapter. Contact BEA Customer Support for additional help.
1027 ELINK_EFML" ERROR : %s	**Finit failed for service %s. Exiting from %s**	

	DESCRIPTION	Adapter attempted to call Finit32() on an FML buffer but received an error.
	ACTION	Verify that buffer sent to adapter was FML32 buffer. Contact BEA Support for additional assistance.
1028 ELINK_ECONFIG" ERROR : %s	**Unable to read next %s section. Exiting from loadConfigurationInformation.**	
	DESCRIPTION	Section of the adapter's configuration file was empty or not present.
	ACTION	Correct configuration file according to format specified in user's guide.
1029 ELINK_ECONFIG" ERROR : %s	**While trying to close tag handle for server section. Exiting from loadConfigurationInformation.**	
	DESCRIPTION	Internal error occurred while reading the adapter configuration file.
	ACTION	Verify that the ADK library (libadk) in use is the original version delivered with the adapter. Contact BEA Customer Support for additional help.
1030 ELINK_EOS" ERROR	**ELINK_EOS No memory allocated by the system for %s. Exiting from loadConfigurationInformation.**	
	DESCRIPTION	Adapter was unable to allocate memory.
	ACTION	Make sure the operating system parameters are set correctly for the amount of memory on the machine and the amount of memory that can be used by a process. Reduce the memory usage on the machine or increase the amount of physical memory on the machine.

1031 ELINK_EFML" ERROR : %s	**Unknown FML field type(Ferror32 = FTYPERR) for %s field in %s. Exiting from %s.**	
	DESCRIPTION	FML32 field is not defined in adapter's environment.
	ACTION	Make sure that FML field tables are specified in adapter's environment file.
1032 ELINK_ECONFIG" ERROR : %s	**Unable to retrieve information about %s group from group hash table. Exiting from %s.**	
	DESCRIPTION	Adapter was unable to retrieve field group information for specified group.
	ACTION	Verify field group is defined in adapter's configuration file.
1033 ELINK_EITYPE" ERROR	**ePA_buildFlist failed for %s service. Exiting from EL_PORTAL_OUT.**	
	DESCRIPTION	Adapter failed to create FLIST for specified service.
	ACTION	Verify FIELD_GROUP definition for specified service is correct. Verify that the version of Portal Infranet is 5.5.3.
1034 ELINK_EOS" ERROR : %s	**Response buffer could not be allocated for %s service. Exiting from EL_PORTAL_OUT.**	
	DESCRIPTION	Adapter was unable to allocate memory for an FML32 response buffer.
	ACTION	Make sure the operating system parameters are set correctly for the amount of memory on the machine and the amount of memory that can be used by a process. Reduce the memory usage on the machine or increase the amount of physical memory on the machine.

1035 ELINK_EFML" ERROR	**Unable to create response FML32 buffer for %s service. Exiting from %s.**	
	DESCRIPTION	Adapter was unable to allocate memory for an FML32 response buffer.
	ACTION	Make sure the operating system parameters are set correctly for the amount of memory on the machine and the amount of memory that can be used by a process. Reduce the memory usage on the machine or increase the amount of physical memory on the machine.
1036 ELINK_ EBUSERR" ERROR	**Business logic error. Exiting from %s.**	
	DESCRIPTION	Error occurred within Portal Infranet while executing specified service.
	ACTION	Verify business logic (account exists for specified request, product exists, etc.) See additional error detail.
1037 ELINK_EFML" ERROR	**While adding %s field for %s group to FML buffer. Exiting from %s.**	
	DESCRIPTION	An error occurred while adding specified FML field to FML buffer.
	ACTION	Verify definition of FML field is correct in FML field table.
1038 ELINK_EAPP_ UNAVAIL" ERROR : %s	**Portal connection could not be made. Exiting from tpsvrinit.**	
	DESCRIPTION	Adapter was unable to establish a connection with Portal Infranet.
	ACTION	Verify Portal Infranet is up. Verify pin.conf file exists in adapter's APPDIR. Verify connection parameters specified in pin.conf are correct.

1039 ELINK_EAPP_ UNAVAIL" ERROR : %s	**Connection to Portal's Infranet could not be made even after three attempts. Exiting tpsvrinit.**	
	DESCRIPTION	Adapter was unable to establish a connection with Portal Infranet.
	ACTION	Verify Portal Infranet is up. Verify pin.conf file exists in adapter's APPDIR. Verify connection parameters specified in pin.conf are correct.
2001 ELINK_ECONFIG" WARNING	**No %s property or property value or invalid value is given. Taking %s default value. In loadConfigurationInformation.**	
	DESCRIPTION	Configuration parameter in adapter configuration file was not specified.
	ACTION	No action necessary.
2002 ELINK_EINVAL" WARNING	**Field %s not present in FML buffer(Ferror32 = FNOTPRES) for %s group. In %s.**	
	DESCRIPTION	FML field is not present in FML buffer.
	ACTION	Correct invoking application to populate FML field in FML buffer. See User's Guide for description of Portal Infranet interfaces and required fields.
2003 ELINK_EOTYPE" WARNING	**%s field not present in %s group in output flist. In %s.**	
	DESCRIPTION	Optional field is not present in FLIST returned from Portal Infranet.
	ACTION	No action is necessary.
2004 ELINK_ECONFIG" WARNING	**Unable to set message levels. Taking default values.**	

	DESCRIPTION	Diagnostic message levels not specified in configration file, adapter used default values.
	ACTION	No action is necessary.
2005 ELINK_EOTYPE" WARNING	**Unknown portal field %s in %s group. In %s.**	
	DESCRIPTION	Field is FLIST is not defined in FIELD_GROUP definition in configuration file.
	ACTION	Define field in configuration file and FML Field table file.
2006 ELINK_EINVAL" WARNING	**Invalid date for %s timestamp field in %s group. In %s.**	
	DESCRIPTION	Date string sent in FML request buffer is not of the format mm/dd/yyyy HH:MM:SS.
	ACTION	Modify invoking application to correctly specify date string.
2007 ELINK_EINVAL" WARNING	**Unknown field %s to FML buffer(Ferror32 = FBADFLD) in %s. In %s.**	
	DESCRIPTION	FML Field is not defined in adapter's environment.
	ACTION	Verify that an FML field definition exists for this field in a FML field table. Verify that the appropriate FML field tables are specified in the adapter's environment file.
2008 ELINK_EINVAL" WARNING	**Invalid occurrence field name %s is given in %s. In %s**	
	DESCRIPTION	Configuration file specifies an occurrence field that is not defined in an FML32 field table.

	ACTION	Make sure that FML32 field tables are specified in adapter's environment. Make sure that FML field is defined for specified occurrence field.
3001 ELINK_INFO" INFO	**Success in adding %s element to %s hash table. In loadConfigurationInformation.**	
	DESCRIPTION	Informational message indicating that configuration information was successfully added to an internal data structure.
	ACTION	No action is necessary.
3002 ELINK_INFO" INFO	**Success in executing %s service.**	
	DESCRIPTION	Informational message indicating that a service executed successfully.
	ACTION	No action is necessary.
	DESCRIPTION	
	ACTION	
3003 ELINK_INFO" INFO	**Success in loading config file.**	
	DESCRIPTION	Informational message indicating that the adapter's configuration file loaded successfully.
	ACTION	No action is necessary.
3004 ELINK_INFO" INFO	**Success in making connection to Portal's Infranet.**	
	DESCRIPTION	Informational message indicating that a connection was established to the Portal Infranet system.
	ACTION	No action is necessary.

3005 ELINK_INFO" INFO	**Success in closing connection to Portal's Infranet.**	
	DESCRIPTION	Informational message indicating that the connection to Portal Infranet was disconnected as part of shutdown processing.
	ACTION	No action is necessary.
3006 ELINK_INFO" INFO	**%s service is advertised.**	
	DESCRIPTION	Informational message indicating that configured service was successfully advertised.
	ACTION	No action is necessary.
3007 ELINK_INFO" INFO	**%s = %s"**	
	DESCRIPTION	Informational message showing configuration parameter and value that was read.
	ACTION	No action is necessary.
3008 ELINK_INFO" INFO	**Entering %s.**	
	DESCRIPTION	Informational message indicating an internal routine was entered.
	ACTION	No action is necessary.
3009 ELINK_INFO" INFO	**Returning from %s.**	
	DESCRIPTION	Informational message indicating that an internal routine was exited.
	ACTION	No action is necessary.
3010 ELINK_INFO" INFO	**EL_PORTAL_OUT called by %s service.**	

	DESCRIPTION	Informational message indicating the name of the service that was invoked.
	ACTION	No action is necessary.
3011 ELINK_INFO" INFO	**Back in %s.**	
	DESCRIPTION	Informational message indicating that control was returned to a internal routine.
	ACTION	No action is necessary.