Britton Lee IDM/Network

Ethernet Controller Board Configuration Manual

(R3v5)

March 1988

Part Number 205-2017-002

This document is specific to XNS and TCP releases of Britton Lee's Ethernet code running on RDBMS Software release 43 and higher.

The information contained within this document is subject to change without notice. Britton Lee assumes no responsibility for any errors that may appear in this document.

The software described in this document is furnished under license and may only be used or copied by the terms of such license.

Table of Contents

Preface

This document describes the procedure for configuring a Britton Lee database management system running either XNS or TCP over an Ethernet controller board.

This document uses Britton Lee Interactive Structured Query Language (Interactive SQL) terminology in describing databases.

This manual provides detailed information on the configuration of XNS and TCP. Users configuring their systems for XNS need not be familiar with the technical aspects of communications protocols. TCP users should be familiar with hexadecimal notation, database administration, and the TCP protocol.

Referenced Documents

This document refers to the following Britton Lee publications:

- *SQL Reference Manual*, Britton Lee part number 205-1344-rev.

  This manual provides an introduction to Britton Lee's Interactive SQL. Users who are installing the XNS protocol and have the Britton Lee utility **idmconfig** need not refer to the *SQL Reference Manual.*

- Britton Lee's *Installation Manual* for your host system.

  This manual describes the installation of Britton Lee's Relational Database Management System for your host computer. It lists the host-dependent configuration steps for the TCP and XNS communications protocols.

1. Ethernet Controller Board Configuration

The Britton Lee Relational Database Management System supports communication via the XNS or TCP protocol suites over a special communications board. The Ethernet controller board will run only one protocol at a time. If you wish to use your controller board for both TCP and XNS communications, you must reconfigure the database server each time you change protocols, or have multiple controller boards.

More than one Ethernet board may be installed in a single database server. The "configure" table must contain configuration information for each.

To use either protocol, you must

- have the Ethernet controller board installed,
- add configuration information to the "configure" table in the "system" database, and
- issue a **reconfigure** command or reboot the database server.

This document describes in detail the configuration information that must be added to the "configure" table. It also shows examples for adding information to the table.

1.1. Configuration Overview

Each Britton Lee database server supports up to four Ethernet controller boards. These boards are numbered from zero to three for each database server. If only one Ethernet board is installed, it is number zero. With two Ethernet boards, they are numbered zero and one. Board number zero corresponds to the board which is in the lowest physical slot number in the database server. Board one corresponds to the board in the higher slot number. For example, if Ethernet boards were in slots 7 and 9, their numbers would be zero and one respectively. If another Ethernet board were placed in slot 6, it would become board zero and the other boards would become one and two respectively.

Each Ethernet board is described by up to eight rows in the "configure" table. "Configure" rows consist of a one-character *type*, a sequence *number*, and a two-byte data *value*. All rows describing the Ethernet controller board are of *type* "E". Throughout the rest of this document, we will refer to *type* "E" rows as E*n*, where *n* is the sequence *number* of the row in question.

The rows are numbered sequentially from zero. E0 through E7 describe the Ethernet board in the lowest-numbered hardware slot. If a second board is installed, it is described by "configure" rows E8 through E15. This numbering scheme continues for each additional board. The board in the next-higher slot number is described by the next group of eight "configure" rows.

1.2. Contents of this Document

Chapter Two describes the configuration rows required for XNS communications.

The first half of Chapter Two is a technical discussion of the configuration information required by XNS. This section will be of interest to system administrators. It will also be useful to users who do not have access to the Britton Lee utility **idmconfig**.

Idmconfig is a special utility for configuring the Britton Lee Relational Database Management System. It is menu-driven, and largely self-documenting. The second half of Chapter Two shows sample **idmconfig** dialogues for XNS configuration.

Users who have **idmconfig**, and are not interested in the details of XNS configuration, should skip directly to section 2.2 to configure their systems with XNS.

Chapter Three covers the configuration rows necessary for communications via TCP. Because **idmconfig** does not yet support TCP configuration, this chapter shows configuration examples as Britton Lee Interactive SQL commands. A TCP configuration checklist appears at the end of Chapter Three.

Chapter Four contains instructions for customers who receive Britton Lee's TCP release. This chapter describes the steps required to load the driver software on the database server. You should complete the procedure described in Chapter Three before beginning this chapter.

2. Configuring XNS

This chapter describes the configuration values required for XNS communications. This introduction, and section 2.1, are fairly technical. Users who have the Britton Lee utility **idmconfig**, and who are not interested in the details of protocol configuration, should skip directly to section 2.2. Section 2.2 shows configuration examples using **idmconfig**.

Figure 1 identifies the individual configuration rows for controller boards using the XNS protocol. Each of these rows is described in greater detail in section 2.1.

XNS Configuration Rows		
Row	*Name*	*Meaning*
E0	Status bits	Defines the protocol to run, specifies whether or not hosts are considered trustworthy, and sets the maximum Level Two data packet size.
E1	Address displacement, LSW	Least significant word of Ethernet address offset.
E2	Address displacement, MSW	Most significant word of Ethernet address offset.
E3	Retry count	Maximum number of times to try retransmitting a packet.
E4	Default identify	The default identify information, including byte and word ordering and character representation, for hosts that do not supply identify packets.
E5	Default hostid	Default host identification number, used when the host fails to send an identify packet.
E6	Hello interval	Number of minutes to wait between checks to see that the host is up.
E7	Retry interval	Interval between attempts to retransmit a packet, in half-seconds.

Figure 1 – XNS Configuration Rows

2.1. Configuring XNS with Interactive SQL

This section describes the contents of the "configure" table for XNS in detail. It also shows configuration examples with Britton Lee's Interactive SQL facility. Users who

have the Britton Lee utility **idmconfig** may be interested in this information, but should use **idmconfig** to configure their systems.

2.1.1. Row E0 – Status Bits

"Configure" row E0 defines the protocol to be run, specifies whether or not the host is considered trustworthy, and sets the maximum Level Two data packet size. Figure 2 shows the status bit fields and describes their contents. Bits 2-10 and 14-15 are reserved for future expansion by Britton Lee.

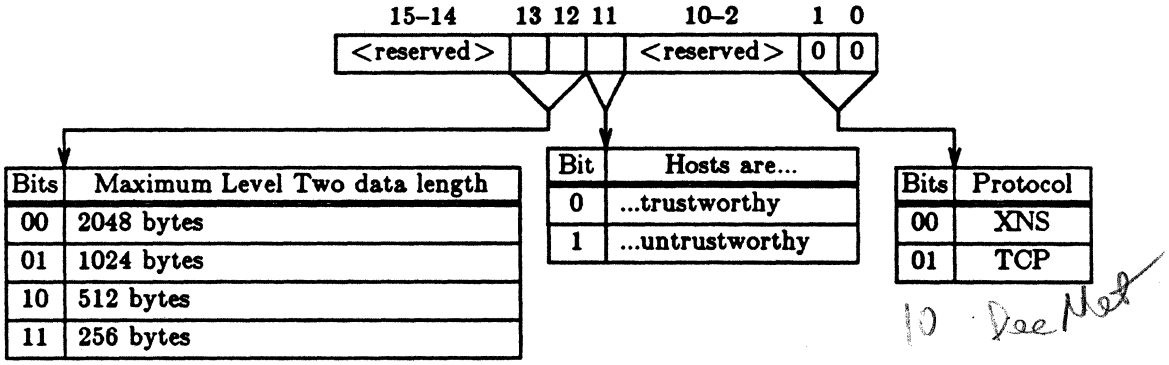

Figure 2 – Row E0 (Status Bits)

Bits 12 and 13, which define the maximum Level Two data packet length, allow values up to 2048 bytes. The standard Ethernet specification does not support values of more than 512 bytes. The higher values will work with Britton Lee software, but may not be supported by third-party vendors.

If you must configure the XNS protocol without using **idmconfig**, use the following procedure. In this example, we assume that all hosts on the network are untrustworthy, that data packets may be up to 1024 bytes in length, and that the protocol is XNS.

(1) Fill in the appropriate bit field values for row E0.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0

(2) Convert this value to hexadecimal. In this case, the value is 0x1800.

(3) Using Interactive SQL, add this "configure" row to the "system" database:

```
1) insert into configure (type, number, value) values ("E", 0, 0x1800);
1 row affected
```

2.1.2. Rows E1 and E2 – Ethernet Address Offsets

Rows E1 and E2 are added to the Ethernet base address to determine the actual Ethernet address of the database server. If you have only one database server on the Ethernet, you are not required to supply values for rows E1 and E2. Britton Lee strongly recommends, however, that you supply a unique Ethernet address for every database server.

2.1.2.1. Calculating a Unique Address Offset

This section explains how to calculate and store a unique Ethernet address for your database server without **idmconfig**. To calculate the address, you must know the serial number of your database server and the number of the Ethernet controller board. See the section Configuration Overview for information on determining the board number for each Ethernet controller board.

A given database server serial number, decomposed into decimal digits, is:

```
OTSSSYMM
```

where O is a real zero, T is is the Britton Lee hardware type (3 for a BL300, 5 for a BL500, etc.), SSS is the true serial number (incremented from 001), and YMM encode the year and the month of manufacture. For example, 05123604 is the 123rd BL500 manfactured, and it was released in April 1986.

If SN is the serial number and CN is the Ethernet controller board number, we can calculate a unique Ethernet address with the following formula.

```
OFFSET = SN / 1000;        integer division (no remainder)
OFFSET = OFFSET * 4;
OFFSET = OFFSET + CN;      add board number to offset
```

Figure 3 – Calculating a Unique Address Offset

OFFSET is the Ethernet address offset for your database server. This number is added to the base Ethernet address, 0x08002c000000, to compute the server's actual Ethernet address. This base address has been assigned to Britton Lee by Xerox Corporation, and is the same for every database system Britton Lee sells.

OFFSET is stored in "configure" rows E1 and E2, as described in the next section.

2.1.2.2. Actual Ethernet Address

The Ethernet address offset OFFSET is stored in two parts. The low word is stored in "configure" row E1, and the high word in row E2. For example, if the value for

OFFSET were 120492:

```
OFFSET = 120492;    (Ethernet address offset, decimal)
OFFSET = 0x1D6AC;   (Ethernet address offset, hexadecimal)

E1 = 0xD6AC;        (low word of OFFSET, hexadecimal)

E2 = 0x0001;        (high word of OFFSET, hexadecimal)
```

Figure 4 – Values of E1 and E2 from Address Offset

The high byte of "configure" row E2 is ignored.

To store the Ethernet address offset on the database server,

1) insert into configure (type, number, value) values ("E", 1, 0xD6AC);
1 row affected

1) insert into configure (type, number, value) values ("E", 2, 0x1);
1 row affected

2.1.2.3. Storing the Ethernet Address on the Host

The Ethernet address must also be stored on each host computer connected to the database server. The location and format of the address is different for different host computer types. For instructions on configuring the host for XNS, see Britton Lee's *Installation Manual* for your host computer system.

2.1.3. Row E3 – Retry Count

"Configure" row E3 determines the maximum number of times Britton Lee's RDBMS communications software will try to retransmit a packet before throwing it away. This should be a small integer value. The default is 15 attempts to retransmit.

This default is appropriate for most situations. To change it to twenty,

1) insert into configure (type, number, value) values ("E", 3, 20);
1 row affected

2.1.4. Row E4 – Default Identify

This "configure" row provides default identification for hosts on the network. It is used only if a host fails to supply an identify packet.

Row E4 describes the host's byte ordering, internal character representation, and data alignment. Like row E0, it consists of bit fields describing attributes of the host. Bits 7-15 are reserved for future expansion by Britton Lee.

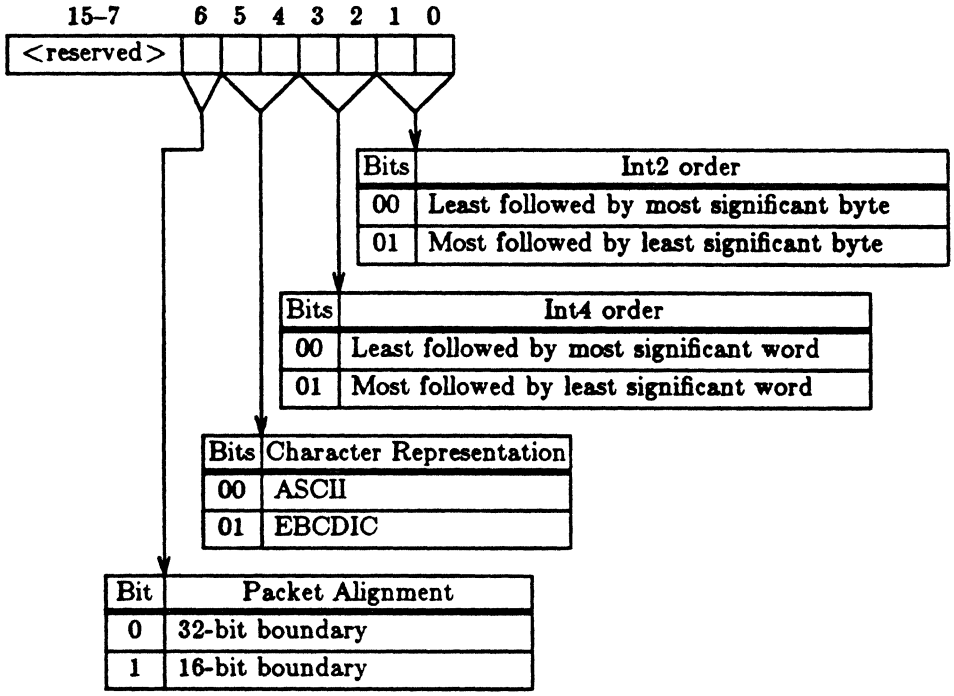

Figure 5 – Row E4 (Default Identify)

If you are unable to use **idmconfig** to configure your Ethernet connection, use the procedure outlined below to set row E4.

In this example, we assume 68000 word and byte ordering, the ASCII character set, and 16-bit packet alignment.

(1) Fill in the bit values for the Default Identify information.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1

(2) Convert this value to hexadecimal. In this case, the value is **0x45 hexa**decimal.

(3) Add the computed value to the "configure" table.

1) insert into configure (type, number, value) values ("E", 4, 0x45);
1 row affected

2.1.5. Row E5 – Default Host ID

Row E5 contains the default hostid. This integer uniquely identifies the host for the database server. Row E5 is used only if the host does not supply an identify packet.

To set the default host identification number to one,

1) insert into configure (type, number, value) values ("E", 5, 1);
1 row affected

2.1.6. Row E6 – Hello Interval

Row E6 is the number of minutes that the database server should wait between checks to see that the host is up. The default is one minute. This default is appropriate for most situations. If you wish to change the interval to two minutes, use the Interactive SQL command

1) insert into configure (type, number, value) values ("E", 6, 2);
1 row affected

2.1.7. Row E7 – Retry Interval

Row E7 specifies the interval between attempts to retransmit a packet. The interval is expressed in half-seconds. The default is one-half second between retransmissions. If you wish to change this default to a full second,

1) insert into configure (type, number, value) values ("E", 7, 2);
1 row affected

2.1.8. Reconfiguring the Database Server

If you are unable to use **idmconfig** for XNS configuration, use the Interactive SQL command **reconfigure**:

```
1) reconfigure;
```

2.2. Configuring XNS with idmconfig

Using **idmconfig** is significantly easier than adding "configure" rows with Interactive SQL commands. This section shows sample **idmconfig** dialogues, and gives suggestions on how best to configure systems using the XNS protocol.

Idmconfig calculates and stores configuration values in the "configure" table of the "system" database. The user makes menu selections appropriate to his system, and **idmconfig** updates the database management system. When all the configuration information is correct, the user may ask **idmconfig** to reconfigure the Britton Lee RDBMS. Reconfiguring makes the new values effective.

Throughout the rest of this chapter, **idmconfig** prompts and output appear in `typewriter` face. User responses appear in **bold**.

2.2.1. Starting idmconfig

To start **idmconfig** on your system, enter the appropriate command from the list below.

UNIX:	**idmconfig**
VMS:	**idmconfig**
VM/CMS:	**idmconfig**
AOS/VS:	**x idmconfig**
PC/MS-DOS:	**IDMCONFIG**

The program begins at the top-level menu. You will see

```
Top Level Menu
1  Display Configure Relation
2  System Configuration Update
3  Communication or Tape Configuration Update
4  Configure the Server With The New Values
5  Exit the program
Select an entry [digit or name] (<CR> to exit this menu):
```

This menu contains five user options which may be selected by name or menu number. Note that all menu items are similarily selected by name or menu number. For convenience, menu names may be abbreviated to the smallest number of characters that avoids ambiguity with other names in the same menu.

In response to the prompt, enter **3**, to change communication configuration.

```
Select an entry [digit or name] (<CR> to exit this menu): 3
```

This brings you to a second-level menu. This menu contains:

```
Update Communication or Tape Configuration
1  Block Mux Configuration
2  Xns Ethernet Interface
3  Parallel Interface (IEEE-488)
4  Serial Interface (RS-232)
5  Magnetic IDM Tape
6  Exit this menu
Select an entry [digit or name] (<CR> to exit this menu):
```

In response to the prompt, enter **2** to configure the database server for XNS communications.

2.2.2. Setting Packet Length and Trustworthy Bits

The XNS protocol requires information on the amount of data that may be transmitted at a single time. The protocol driver must also know whether or not hosts on the network are considered trustworthy.

The following example shows an **idmconfig** dialogue that sets the data packet size to 2048 bytes and declares all hosts trustworthy.

```
XNS Ethernet Configuration
1  Packet Size and Trustworthy Bits
2  Address
3  Host Characteristics
4  Retry Count [default 15]
5  Retry Interval [default 1]
6  Hello Interval [default 1]
7  Host Identification [default -1]
8  Exit this menu
Select an entry [digit or name] (<CR> to exit this menu): 1
```

Option **1** controls "configure" row E0. This row includes the maximum Level Two data packet size, the bits that define the trustworthiness of the host, and the protocol to run. The protocol is XNS, as selected in the previous menu.

```
Ethernet Status
1  Set Trustworthy Bits
2  Set Packet Size
3  Update Server and Exit this menu
4  Exit this menu
Select an entry [digit or name] (<CR> to exit this menu): 1
```

Option **1** prints a sub-menu that allows us to choose trustworthy or non-trustworthy hosts.

```
Set Trustworthy Parameters
1  Trustworthy Host User Names
2  Untrustworthy Host user ids
3  Exit this menu
Select an entry [digit or name] (<CR> to exit this menu): 1
```

Option **1** declares that all host computers connected to the database server are trustworthy. We then return, automatically, to the previous menu.

```
Ethernet Status
1  Set Trustworthy Bits
2  Set Packet Size
3  Update Server and Exit this menu
4  Exit this menu
Select an entry [digit or name] (<CR> to exit this menu): 2
```

Option **2** prints a sub-menu that allows us to choose the maximum length of the Level Two data packet.

```
Channel Packet Sizes
1  2K bytes
2  1K bytes [default]
3  512 bytes
4  256 bytes
5  Exit this menu
Select an entry [digit or name] (<CR> to exit this menu): 1
```

We choose the largest possible size. **Idmconfig** returns us to the previous menu.

```
Ethernet Status
1  Set Trustworthy Bits
2  Set Packet Size
3  Update Server and Exit this menu
4  Exit this menu
Select an entry [digit or name] (<CR> to exit this menu): 3
```

The **3** entered in response to the last menu prompt saves our configuration values on the database server.

2.2.3. Assigning an Ethernet Address

Every device using the XNS protocol on an Ethernet must have a unique Ethernet address. The following dialogue shows how to assign a unique Ethernet address to your database server. You must know the database server's serial number and the number of your Ethernet controller board in order to assign the address. See the section Configuration Overview for information on determining Ethernet controller board numbers.

We assume that you are at the "XNS Ethernet Configuration" sub-menu of **idmconfig**.

```
XNS Ethernet Configuration
1  Packet Size and Trustworthy Bits
2  Address
3  Host Characteristics
4  Retry Count [default 15]
5  Retry Interval [default 1]
6  Hello Interval [default 1]
7  Host Identification [default -1]
8  Exit this menu
Select an entry [digit or name] (<CR> to exit this menu): 2
```

Option **2** selects the submenu to set the Ethernet address for the database server.

```
Ethernet Address
1  Build XNS Address from Serial Number (Recommended by BLI)
2  Set XNS Address by Number
3  Look up XNS Address in XNSHOSTS System File
4  Exit this menu
Select an entry [digit or name] (<CR> to exit this menu): 1
```

Option **1** causes **idmconfig** to compute a unique Ethernet address from the serial number of the database server. You will be prompted for the server's serial number, and for the number of the Ethernet controller board.

```
Enter Serial Number of your Database Server:  05126611
Enter Ethernet Card Number [0, 1, 2, 3]: 0
Please write down the following value to be added to your
system xnshosts file
0.0.0.1:8.0.44.0.80.12
```

For each of the prompts above, you should enter the appropriate value for your hardware.

The last line of the dialogue supplies you with the Ethernet address for your database server. You must edit your host system services file to include the Ethernet address. The method for doing so depends on the host system you are running. Refer to Britton Lee's *Installation Manual* for your system for instructons.

2.2.4. Setting the Retry Count

The retry count is the number of times that the RDBMS software should try to retransmit a packet to the host, in the event of transmission errors. The default is 15 retries. This value is appropriate for most networks.

This value may be changed from the "XNS Ethernet Configuration" menu of **idmconfig**:

```
XNS Ethernet Configuration
1  Packet Size and Trustworthy Bits
2  Address
3  Host Characteristics
4  Retry Count [default 15]
5  Retry Interval [default 1]
6  Hello Interval [default 1]
7  Host Identification [default -1]
8  Exit this menu
Select an entry [digit or name] (<CR> to exit this menu): 4
Enter retry count: 20
```

changes the retry count to 20.

2.2.5. Setting Default Identify Information with idmconfig

This section shows an **idmconfig** session for setting the default host identification information. We assume that the user is at the "XNS Ethernet Configuration" menu of **idmconfig**.

```
XNS Ethernet Configuration
1  Packet Size and Trustworthy Bits
2  Address
3  Host Characteristics
4  Retry Count [default 15]
5  Retry Interval [default 1]
6  Hello Interval        [default 1]
7  Host Identification [default -1]
8  Exit this menu
Select an entry [digit or name] (<CR> to exit this menu): 3
```

Option **3** sets the value of "configure" row E4, the Default Identify row.

```
Ethernet Host Characteristics
1  Use This Host's Characteristics
2  Set Explicit Host Characteristics
3  Exit this menu
Select an entry [digit or name] (<CR> to exit this menu): 2
```

Option **2** tells **idmconfig** that we wish to set the default characteristics explicitly. If we had entered **1**, the characteristics of the host on which **idmconfig** is running would be used. Using the characteristics of the current host is acceptable in most cases.

```
Ethernet Host Characteristics
1  Ascii Character Set
2  Ebcdic Character Set
3  Least Before Most Significant Byte (eg. VAX)
4  Most Before Least Significant Byte (eg. 68000)
5  Least Before Most Significant Word (eg. VAX)
6  Most Before Least Significant Word (eg. 68000)
7  Update Server and Exit this menu
8  Abort this menu
Select an entry [digit or name] (<CR> to exit this menu): 1
```

Use the ASCII character set.

```
Ethernet Host Characteristics
1  Ascii Character Set
2  Ebcdic Character Set
3  Least Before Most Significant Byte (eg. VAX)
4  Most Before Least Significant Byte (eg. 68000)
5  Least Before Most Significant Word (eg. VAX)
6  Most Before Least Significant Word (eg. 68000)
7  Update Server and Exit this menu
8  Abort this menu
Select an entry [digit or name] (<CR> to exit this menu): 3
```

Use VAX byte ordering.

```
Ethernet Host Characteristics
1  Ascii Character Set
2  Ebcdic Character Set
3  Least Before Most Significant Byte (eg. VAX)
4  Most Before Least Significant Byte (eg. 68000)
5  Least Before Most Significant Word (eg. VAX)
6  Most Before Least Significant Word (eg. 68000)
7  Update Server and Exit this menu
8  Abort this menu
Select an entry [digit or name] (<CR> to exit this menu): 5
```

Use VAX word ordering.

2.2.6. Setting the Default Host ID

The default host id is used by the RDBMS software if the host fails to send an identify packet. If defined, it must be the same as the IDMHOSTID parameter in the system parameter file.

This value may be set from the "XNS Ethernet Configuration" menu of **idmconfig**.

2.2.7. Setting the Hello Interval

The hello interval is the amount of time, in minutes, that the RDBMS software should wait between handshakes with the host. The default is one minute.

This value may be changed from the "XNS Ethernet Configuration" menu of **idmconfig**.

2.2.8. Setting the Retry Interval

Row E7 defines the amount of time that the RDBMS software should wait between attempts to retransmit a packet to the host, in case of transmission problems. The interval is expressed in half-seconds. The default is one-half second.

This value may be changed from the "XNS Ethernet Configuration" menu of **idmconfig**.

2.2.9. Reconfiguring the Database Server

Once all the "configure" values for the Ethernet XNS protocol have been added, you must issue a **reconfigure** command or reboot the database server.

To issue a **reconfigure** command from **idmconfig:**

```
Top Level Menu
1  Display Configure Relation
2  System Configuration Update
3  Communication or Tape Configuration Update
4  Configure the Server With The New Values
5  Exit the program
Select an entry [digit or name] (<CR> to exit this menu): 4
```

Option **4** puts the new values into effect. You have now completed XNS configuration.

3. Configuring TCP

This chapter, and Chapter Four, describe the procedure for configuring your system for TCP communications. To aid in understanding the TCP configuration, some orientation is in order.

Using the TCP protocol to communicate with your database server requires three things:

(1) TCP must be supported in the kernel on the host computer.

(2) The "configure" table in the "system" database must be set up for TCP.

(3) Britton Lee's TCP channel software must be installed on the database server.

To determine whether or not TCP is supported on your host (requirement (1)), type the command

rlogin localhost

at the Unix or Ultrix shell prompt. If the rlogin succeeds, TCP is installed, and you may log out. If it does not succeed, TCP is not supported on your system, and you cannot use it to communicate with the database server.

The rest of this chapter describes the procedure for satisfying requirement (2). It lists all the steps necessary for configuring your database server for TCP. Where possible, this document supplies the values that you should store in the database. If no default is appropriate, instructions for calculating a value are provided. At the end of this chapter, a checklist summarizes the steps required.

Chapter Four contains instructions for loading the TCP driver on the database server. It satisfies requirement (3).

You should read this chapter and Chapter Four completely before you begin the configuration. When you have read both chapters, return to the beginning of Chapter Three. Carry out all the steps listed. Then follow the instructions in Chapter Four.

If you have problems during the configuration, please call Britton Lee Customer Support.

3.1. TCP Configuration Rows

This section lists the rows in the "configure" table of the "system" database that are required for TCP configuration.

TCP configuration rows have a one-character *type*, a sequence *number*, and a *value*. All Ethernet rows have *type* "E". Throughout this document, we will refer to these rows

with the abbreviation E*n*, where *n* is the sequence *number* of the row in question.

Eight "configure" rows are used for each Ethernet board when configuring TCP. Five are required(E0, E1, E2, E5, and E6). The rest have default values. With multiple Ethernet controller boards, each board is associated with a set of "E" rows. Refer to section 1.1 for information on this how the rows are associated with each board. The contents of each row are shown in the table below.

TCP Configuration Rows		
Row	*Name*	*Meaning*
E0	Status bits	Specifies whether or not hosts are considered trustworthy, sets the maximum Level Two data packet size and sets the type of protocol to be used (XNS or TCP).
E1	Address displacement, LSW	Least significant word of Ethernet address offset. This row is identical to E1 for XNS.
E2	Address displacement, MSW	Most significant word of Ethernet address offset. This row is identical to E2 for XNS.
E3	Retry Count	Maximum number of times to try retransmitting a packet.
E4	TCP Status	Unused.
E5	Internet address, LSW	Least significant word of the Internet address.
E6	Internet address, MSW	Most significant word of the Internet address.
E7	Retry interval	Interval between attempts to retransmit a packet, in half-seconds. This row is identical to E7 for XNS.

Figure 6 – TCP Configuration Rows

These rows are described in greater detail below. The Britton Lee utility **idmconfig** does not yet support TCP configuration. Instead, in this section we show sample SQL commands to add information to the "configure" table. For more information on SQL, see *SQL Reference Manual*, Britton Lee part number 205-1344-rev.

3.2. Row E0 – Status Bits

"Configure" row E0 defines the trustworthiness of hosts on the network, sets the maximum Level Two data packet size and determines the protocol to run (XNS or TCP). It contains bit fields that store this information.

The bit fields, and their legal values, are shown in figure 7, below.

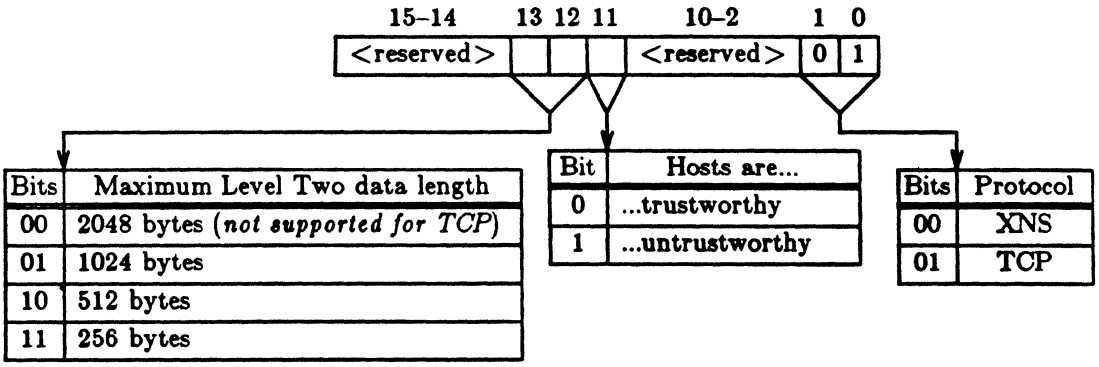

Figure 7 – Row E0 (Status Bits)

Bits 0 and 1 are unused unless your Britton Lee database server is running IDM/RDBMS 43C or higher software.

Bits 12 and 13 define the maximum Level Two data packet size. Britton Lee software does not support packet sizes above 1024 bytes under the TCP protocol.

Choose the appropriate value for your system from the table below.

Packet size	*Host status*	*use the value...*
512 bytes	trustworthy	8193
	untrustworthy	10241
1024 bytes	trustworthy	4097
	untrustworthy	6145
256 bytes	trustworthy	12289
	untrustworthy	14337

Add the value chosen above with the Interactive SQL command

1) insert into configure (type, number, value) values ("E", 0, *chosen_value*);
1 row affected

3.3. Rows E1 and E2 – Ethernet Address Offsets

Rows E1 and E2 supply the address offset for Ethernet address calculation. They are identical to those described for XNS installations.

The Ethernet address is the address of the database server on the Ethernet. If you have only one database server on your Ethernet, you are not required to supply values for rows E1 and E2. Britton Lee strongly recommends, however, that every database server be assigned a unique Ethernet address.

Below, we describe how to calculate a unique Ethernet address from the serial number of the database server and the number of the communications controller board.

3.3.1. Calculating a Unique Address Offset

Every Ethernet controller board must have a unique Ethernet address. This section describes how to calculate a unique address, given the serial number of the database server and the number of the controller board. See the section Configuration Overview for information on determining the Ethernet controller board number.

A given database server serial number, decomposed into decimal digits, is

```
OTSSSYMM
```

where O is a real zero, T is the server type (3 for a BL300, 5 for a BL500, 7 for a BL700, etc.), SSS is the true serial number (incremented from 001), and YMM encode the year and the month of manufacture. For example, 05123604 is the 123rd BL500 manfactured, and it was released in April 1986.

If SN is the serial number and CN is the board number:

```
OFFSET = SN / 1000;     integer division (no remainder)
OFFSET = OFFSET * 4;
OFFSET = OFFSET + CN;   get unique address for each board
```

Figure 8 – Calculating a Unique Address Offset

OFFSET is a unique Ethernet address offset for your database server.

3.3.2. Actual Ethernet Address

The actual Ethernet address is calculated according to the following formula:

```
  0x08002c000000  (Ethernet base address, in hexadecimal)
+         OFFSET  (Ethernet address offset, from the last section)
  --------------
= Actual Ethernet Address
```

Figure 9 – Calculating the Actual Ethernet Address

The database server will calculate the actual Ethernet address, using the formula in Figure 9. The base address 0x08002c000000 has been assigned to Britton Lee by Xerox Corp., and is the same for every system Britton Lee sells. The Ethernet address offset (from figure 8) must be stored in "configure" rows E1 and E2.

Row E1 contains the least-significant word of the address offset. Row E2 contains the most-significant word. For example, if the value of OFFSET from figure 8 were 120492, then:

```
OFFSET = 120492;    (Ethernet address offset, decimal)
OFFSET = 0x1D6AC;   (Address offset, hexadecimal)

E1 = 0xD6AC;        (Row E1 - low-order word of OFFSET, hexadecimal)

E2 = 0x01;          (Row E2 - high-order word of OFFSET, hexadecimal)
```

Figure 10 – Values of E1 and E2 from Address Offset

The high-order byte of E2 is ignored.

To add these rows to the "configure" table, we use the Interactive SQL commands

```
1) insert into configure (type, number, value) values ("E", 1, 0xD6AC);
1 row affected

1) insert into configure (type, number, value) values ("E", 2, 0x1);
1 row affected
```

It is not necessary to store the Ethernet address on the host for TCP communications.

3.4. Row E3 – Retry Count

"Configure" row E3 supplies the retry count for network communications. If the host receives a bad packet, the communications software will attempt to retransmit it. This row sets an upper limit for retransmissions.

You should use 100. This is appropriate for most installations. To set this value,

```
1) insert into configure (type, number, value) values ("E", 3, 100);
1 row affected
```

3.5. Row E4 – TCP Status

This row is currently unused and can be ignored. It should not be present in the "configure" table.

3.6. Rows E5 and E6 – Internet Address

Internet addresses are four-byte values. Rows E5 and E6 are concatenated to provide the address for database server.

The database server listens to data transmitted on the Ethernet. If it recognizes a packet intended for its Internet address, the database server acknowledges and processes it. Both the host and the database server must agree on this Internet address and it must be unique.

There is no default Internet address for TCP. You must supply one when you configure the Ethernet controller board. To do so, follow the procedure outlined below.

(1) Find the Internet address for your server. On most Unix systems, you may find this address in the file /etc/hosts. The address will be four numeric values, separated by periods. For more information on the Internet address, refer to Britton Lee's *Host Software Installation Manual* for your system.

(2) Convert each period-separated number in the address to hexadecimal. Most scientific calculators can do this. If you do not have one available, you may use the Unix program *dc* as follows:

   (a) Assume that the Internet address for this server is 192.23.25.1. We wish to convert 192, 23, 25, and 1 into hexadecimal numbers.

(b) Type *dc* at the shell prompt on your Unix system.

(c) Type **16o** (sixteen-oh, not zero). This instructs the *dc* program to output all numbers in base 16.

(d) Type the first number you want converted, followed by a lower-case "p":

```
192p
C0
```

Dc converted 192 decimal into C0 hexadecimal.

(e) Repeat Step (d) for each of the period-separated numbers in the address.

(3) When all parts of the address have been converted, write them down in two-byte pairs. In our example,

192	23	25	1

is

C0	17	19	01

(4) Add the hexadecimal values to "configure" rows E5 and E6. Note that the most significant two bytes (0xC0 and 0x17) are stored in row E6, and the least significant bytes are stored in E5.

```
1) insert into configure (type, number, value) values ("E", 5, 0x1901);
1 row affected

1) insert into configure (type, number, value) values ("E", 6, 0xC017);
1 row affected
```

If both **inserts** succeeded, you may skip to section 3.7. The rest of this section explains how to deal with errors generated during the steps above.

If either of the **insert** commands above generates an "overflow" error, the value is too large to fit into a "configure" row as a positive number. It is still possible to add the value to the "configure" table. Use the method described below.

(1) Convert the number that overflowed from hexadecimal to decimal. For example, if 0xC017 is too large, convert it to decimal. To do this is *dc*:

(a) Type *dc* in response to the Unix or Ultrix shell prompt.

(b) Type **16i**, which instructs *dc* to interpret input in base 16.

(c) Type the number you want converted, followed by a lower-case "p":

```
C017p
49175
```

Dc converted C017 hexadecimal to 49175 decimal.

(2) Clear the sign bit of the decimal number. To do this, subtract the number from 65536:

```
  65536
- 49175
  -----
  16361
```

This number is small enough to add to the "configure" table, but you must first re-set the sign bit by negating the number. In our example, 16361 becomes −16361.

(3) Add this value to the "configure" table:

```
1) insert into configure (name, type, value) values ("E", 6, -16361);
1 row affected.
```

3.7. Row E7 – Retry Interval

Row E7 specifies the interval between attempts to retransmit a packet. The interval is given in halves of a second. A two-second delay is recommended. To set a two-second delay, set row E7 to 4:

```
1) insert into configure (type, number, value) values ("E", 7, 4);
1 row affected
```

3.8. Configuration Completed

When you reach this point, you have completed the configuration of TCP. You must now load the TCP channel software onto the database server. For details on doing so, see Chapter Four.

3.9. TCP Configuration Checklist

A configuration checklist is provided below.

- ☐ Set TCP status bits.
  - ☐ Find the value appropriate to your system in the table in Section 3.2.
  - ☐ Add the value to "configure" row E0.
- ☐ Set Ethernet address offset.
  - ☐ Calculate address offset according to the formula given in Section 3.3.1.
  - ☐ Convert the offset to hexadecimal. See Section 3.6 for hints on how to do decimal/hexadecimal conversion.
  - ☐ Split the address into its high and low parts, as described in Section 3.3.2. Add them to the "configure" table in rows E1 and E2.
- ☐ Set the retry count, row E3, to 100.
- ☐ Row E4 is unused and should be ignored. It should not appear in the "configure" table.
- ☐ Set the database server's Internet address.
  - ☐ Convert the Internet address to hexadecimal, using the procedure described in Section 3.6.
  - ☐ Split the address into its upper and lower halves.
  - ☐ Add these to "configure" rows E5 and E6.
- ☐ Set the retry interval for communications to 2 seconds. To do so, set row E7 to 4.

---

4. Loading the TCP Driver on the Server

If your system is running IDM/RDBMS release 43C or higher, disregard this section. With 43C and higher, the default Ethernet code (XNS, TCP, etc.) to run is determined by setting bits 0 and 1 in row E0 of the "configure" table. This procedure is described in the section TCP Configuration Rows.

If you are running an IDM/RDBMS release lower than 43C and you have completed the procedure described in Chapter Three, you must now load the TCP channel software on your database server. The tape included with this distribution contains the TCP Ethernet channel code.

Release 43 of Britton Lee's RDBMS software automatically loads and executes a program called "ethernet" when the server is booted. The "ethernet" file distributed with Release 43 implements the XNS protocol. To use the TCP protocol, the database server must execute the code supplied on the distribution tape instead.

There are two choices for installing the TCP channel code. You may either

(a) install the software so that it is automatically loaded when you reboot, or

(b) continue to use the XNS code by default, but make the TCP code available for manual loading.

Option (a) is appropriate if you wish to switch your normal communications protocol to TCP. Option (b) is useful if you will continue to use the XNS interface most of the time. If you choose option (b), however, you must reconfigure the database server every time you change protocols.

If you want to install TCP so that it is loaded automatically every time you reboot (option (a)), follow the procedure outlined in Section 4.1. If you wish to continue using XNS by default, follow the instructions in Section 4.2.

4.1. Installing TCP as the Default Driver

To load the TCP Channel code automatically every time you reboot, you must store it in a file called "ethernet". The code in that file is loaded automatically every time you boot.

This requires loading the code called "tcp" from tape, moving the old "ethernet" code to a new file, and changing the name of the "tcp" code to "ethernet".

The procedure for doing so is described below.

4.1.1. Load the Software from Tape

Mount the supplied tape on the tape drive. On the database server console, type the command

```
kernal fileload -t tcp
```

This will create a file called "tcp" on the server.

4.1.2. Move the XNS Code

Change the name of the XNS code from "ethernet" to "xns". To do this in SQL, type the command

```
1) update relation set name = 'xns' where name = 'ethernet';
1 row affected
```

4.1.3. Move the TCP Code

Now change the name of the TCP channel code from "tcp" to "ethernet".

```
1) update relation set name = 'ethernet' where name = 'tcp';
1 row affected
```

4.1.4. Save Your Changes

To be sure your changes are written to disk, type the Britton Lee SQL command

```
1) sync;
```

The TCP code will now be loaded automatically every time you reboot.

4.1.5. Reboot the Database Server

To make the TCP configuration changes effective, reboot your database server. You have now completed the installation of TCP.

4.2. Using XNS as the Default Driver

If you wish to continue to use XNS as your default driver, you may load the TCP driver into another file and execute it when you need it. This requires that you change the "configure" table and reboot the server every time you want to change protocols.

To do this, you must load the software from tape. When you want to run TCP, you must manually invoke it from the database server's console.

4.2.1. Loading the Software from Tape

Mount the supplied tape on the tape drive. On the database server console, type the command

kernal fileload

This will create a file called "tcp" on the server.

4.2.2. Using the TCP Driver

Every time you want to run TCP, you must change the "configure" table according to the instructions in Chapter Three, and reboot your database server.

When you reboot, move the server's mode switch to the MAINT position and type

kernal –g*n*tcp

where *n* is the slot number that contains the Ethernet controller board. This overrides the default communications software, and uses TCP instead.

Now move the switch to the RUN position.

4.2.3. Using the XNS Driver

When you want to use the XNS driver, you must reconfigure your database server according to the instructions in Chapter Two. When you have completed the

configuration, reboot the server. The XNS driver code will be loaded automatically.