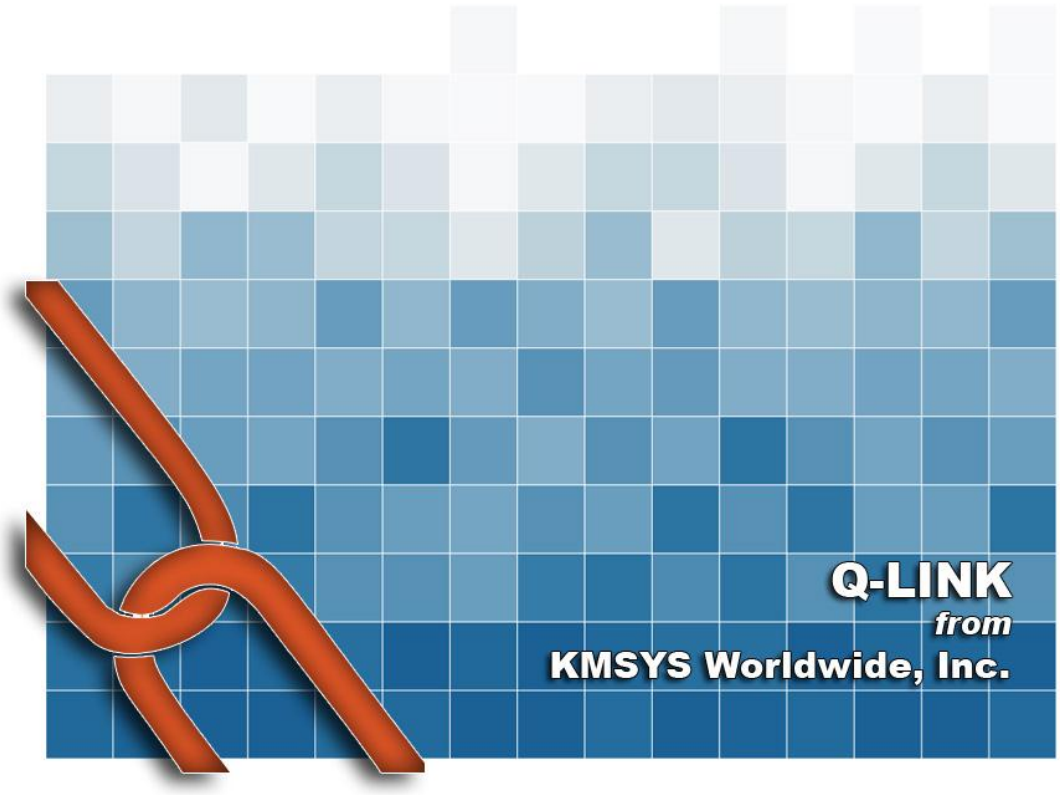


KMSYS
WORLDWIDE, INC

Installation and Operation Guide

10 August 2011

The information contained in this document is the latest available at the time of preparation; therefore, it may be changed without notice, and it does not represent a commitment on the part of KMSYS Worldwide, Inc. The software described in this document is furnished under a license agreement. The software may be used or copied only in accordance with the terms of the agreement. It is against the law to copy this software on magnetic tape, disk, or any other medium for any purpose other than stated in the terms of the agreement, or without the express written permission of KMSYS Worldwide, Inc.

If you have any comments about the software or documentation, notify KMSYS Worldwide, Inc., in writing at the following address:

KMSYS Worldwide, Inc.
P.O. Box 669695
Marietta, Georgia 30066
U.S.A.

Technical Support (770) 635-6363 - Main Number (770) 635-6350 - Fax (770) 635-6351

Q-LINK Release 6R6, November 1999

Table of Contents

Important Notices!

For sites who are evaluating Q-LINK for possible purchase, please see Section 1.6, "Quick Installation".

For those sites that are migrating from earlier release levels of Q-LINK, please read the Q-LINK README.TXT file on the Documentation CD from KMSYS Worldwide prior to installing Q-LINK. Some of the differences described in that document may effect the Q-LINK operating environment at your site.

Any Q-LINK programs in object form, SAVEed with previous levels of Q-LINK, ***must*** be recompiled to execute under this version. KMSYS Worldwide, Inc. provides an SSG routine called "COMPILE-RUNS" which will automatically recompile all Q-LINK programs. This routine can be found in the second Q-LINK product file, SYSLIB*QLINK-1 (default mode install). For other modes of installation, refer to the COMUS System Registration Log (SRL).

References are made throughout this document to two KMSYS Worldwide products: Q-LINK and InfoQuest. This guide can be used to install Q-LINK for both products. Any reference to Q-LINK in conjunction with InfoQuest should be ignored by Q-LINK-only users.

If you are installing Q-LINK in preparation for installing InfoQuest, please see "Minimum Configuration Requirements" in Chapter 1, Section 1.5, for the Q-LINK sizing information required for InfoQuest operation.

With DMS 2200, level 9R1 or higher, the term, "DMR", is no longer used. Instead, Unisys documentation now refers to a logical data manager (LDM) for DMS running under Universal Data System (UDS) Control. Q-LINK can interface with any number of LDMs through UDS Control; however, this documentation will continue to use the term, "DMR", in a generic fashion since Q-LINK can interface with non-UDS DMS applications (e.g., 8R3) as well as DMS applications running under UDS Control.

Chapter 1: General Information

This manual contains technical information regarding the installation, support and operation of Q-LINK, and is directed to systems support personnel.

For sites that are evaluating Q-LINK for possible purchase, please see Section 1.6, "Quick Installation."

1.1 Overview of Chapters

The manual is divided into seven major chapters:

- Chapter 1 provides notes concerning the use of this publication, lists supporting software levels and minimum requirements, describes the release tape contents, and provides a quick installation procedure.
- Chapter 2 contains important information that should be considered prior to generating/installing Q-LINK.
- Chapter 3 shows the steps necessary to generate and install the Q-LINK.
- Chapter 4 illustrates a simple procedure to verify the Q-LINK installation.
- Chapter 5 discusses two possible Q-LINK configurations for 2200/2200 systems.
- Chapter 6 provides a detailed description of the Q-LINK security facility and each SGS necessary to implement security.
- Chapter 7 provides a technical description of the Q-LINK/BIS interface, important considerations and a guide to operations. This chapter should be referred to for system start-up and day-to-day operations.
- Chapter 8 provides a guide to the operations of the Q-LINK DDP-PPC interface that allows the execution of Q-LINK programs requested on one host and executed on a second host.
- The remaining chapters contain considerations when installing and applying changes to subsequent installs.

1.2 Additional Documentation

The following manuals are available with the release of Q-LINK:

- Q-LINK Applications Development User Guide
- Q-LINK Programmer Reference

The User Guide contains many examples illustrating Q-LINK being used for BIS applications development, prototyping, program testing and restructuring.

1.3 Syntax Notation

The following conventions are used throughout this manual in the description of Q-LINK commands:

- Changes to this document since its last publication are marked with a change bar (an elongated vertical bar) as shown to the right of this paragraph.
- Important notes and warnings are encased in a box as shown around this bullet.
- All words in UPPERCASE letters (not italicized) are reserved keywords and must be entered exactly as shown.
- All italicized words (mostly in lowercase letters) are to be substituted by a user supplied name or value.
- Ellipsis (…) implies allowable, but omitted, repetitions in the published syntax. Please note that the ellipsis is ***not*** allowed in the command or directive when parsed.
- A vertical bar (|) represents an "or" or "and/or" operator.
- Selections appearing within brackets, "[ ]", are lists of optional items of which one may be selected. In the following example, neither A nor B is required, but either one or the other may be selected:

  [A | B]
- An underlined word in optional brackets represents the default value when not entered.
- Selections appearing within braces, "{ }", are lists of items of which one and only one must be selected. In the following example, one of either C, D or E must be selected:

  {C | D | E}
- Selections appearing within double vertical bars, "||", are lists of items of which one or more may be selected. The items between vertical bars are referred to as permutations and may be selected in any order. In the following example, one or more of F, G, and/or H must be selected:

  ||F | G | H||

1.4 Supporting Software Levels

Any supported level of the following Unisys OS 2200 software products may be used for the installation and operation of Q-LINK:

ASCII COBOL
COMUS
ED
ELT
EXEC
MAP
BIS
MASM
PCIOS
PLUS
SORT
SSG
SYSLIB
UCSRTS
Or, any supported OS 2200 base release.

A number of issues exist when interfacing with certain levels of OS 2200 software. There is a discussion of these problems in Chapter 2, "Installation Guidelines". In addition, upcoming levels of software may require additional considerations when running Q-LINK Level 6. KMSYS Worldwide may issue additional release tapes for Level 6 without reprinting this Installation Guide. Any changes and/or considerations after the publication of this document may be found in the first file (Q0) of the release tape. Please read the element INFO/QLINK6R1 prior to installing Q-LINK. The INFO/QLINK6R1 element will be displayed (REGISTER,S) or @SYMed by COMUS during product registration.

Any currently supported levels of the following optional Unisys OS 2200 software products are only required for the installation and operation of Q-LINK if a particular file system interface is desired:

DMS 2200 Interface:	DMS 2200
DTM Interface:	BIS
RDMS 2200 Interface:	RDMS 2200
SFS 2200 Interface:	SFS 2200
DDP-PPC Interface:	DDP-PPC

This level of Q-LINK supports remote Q-LINK execution. The remote execution facility allows Q-LINK on one system (requester) to pass a request to Q-LINK on another system (executor). The executor system will process the request using data available on the local system, and pass the result back to the requester system. In order for Q-LINK to support remote execution, Q-LINK must be generated with the DDP-PPC facility. The DDP-PPC facility has several prerequisites, which are detailed in the Unisys publication entitled Distributed Data Processing Implementation and Administration Guide (3787-3270-000). The Q-LINK Installation Guide will only address the steps required to allow Q-LINK to operate in an existing DDP-PPC environment (see Chapter 5, "Q-LINK Configurations" and Chapter 8, "Q-LINK DDP Operations").

1.5 Minimum Configuration Requirements

This subsection only applies when installing Q-LINK to support InfoQuest. The minimum configuration requirements shown are established by KMSYS Worldwide in order to generate the InfoQuest programs.

The Q-LINK System is an integral part of InfoQuest and must be configured accordingly. The following minimum configuration values must be set on the configuration screen, "Basic Parameters for Q-LINK Generation (SCREEN: BUILD)", shown below:

```
                                                              SCREEN: BUILD
These parameters define information which can only be changed by a Q-LINK
system generation. Enter a question mark (?) in any field and transmit for
information on how to configure that parameter. DMS1100 related parameters
can be viewed with a selection from the main menu (use the BACK command).


MAX VARIABLE NAMES                          400
DATA STORAGE AREA (DOUBLE WORDS)            1000
PROGRAM STORAGE SIZE (COMMANDS)             1400
MAXIMUM LABELS IN A PROGRAM                 400
MAXIMUM FILES (DEF F DIRECTIVES)            10

Q-LINK DBANK START ADDRESS                  0154000
DB4 PRODUCT LIBRARY FILE                    NONE
DTM PROC FILE
DTM SCHDLR INTERFACE BDI                    0403133
QWIZZ PRODUCT FILE                          NONE
DDP-PPC RELOCATABLE FILE                    NONE
DDP-PPC PROC FILE                           NONE
```

If you choose to generate Q-LINK at your site, these minimum configuration values (shown above) must be set on the Q-LINK configuration screen, "Basic Parameters for Q-LINK Generation (SCREEN: BUILD)", prior to starting the COMUS BUILD runstream for Q-LINK.

PROGRAM STORAGE SIZE = 1800 is only required for the server used to generate (compile) the InfoQuest programs. The server used for report creation could be from a Q-LINK generation configured for 1200 commands or less.

InfoQuest programs use a minimum of three dummy PCIOS file definitions internally. While these do not relate to actual files, they reduce the number of PCIOS files available to the user. If any DBM will access more than seven user PCIOS files, the parameter, "MAXIMUM FILES (DEF F DIRECTIVES)", should be increased. Note: Increasing the maximum number of files does not increase the file table space appreciably.

InfoQuest requires a minimum RDA size of 4,000 words to function properly. This parameter is a dynamic configuration parameter (not shown here) found on the continuation screen for the "Basic Q-LINK Dynamic Configuration Parameters". Its default as released by KMSYS Worldwide is 4,000 words but may be increased as required.

When InfoQuest is first installed, Q-LINK must be configured with a "PROD1" server class (see Section 3.6.1). Note: When Q-LINK is initially installed, it is already configured with a batch server named PROD1; however, if this name is changed initial access to InfoQuest will fail.

All remaining Q-LINK configuration parameters will depend on individual local site requirements.

1.6 Release Tape Contents

The following files are included on the Q-LINK release tape:

File	Description
0	This is the COMUS utility file.
1	This file contains all base symbolic elements and fixed relocatable elements needed to generate Q-LINK.
2	This file contains any runstreams used in non-COMUS system generation and configuration.
3	This is the Q-LINK permanent change file in program file format. It is used to apply changes to the base symbolic elements in file 2.
4	This file is initially empty. It will contain the generated relocatables, PROCs and MAP symbolics output from the generation process.
5	This file contains an initial Q-LINK configuration generated at KMSYS Worldwide (see Section 1.6, "Quick Installation", page 1-8). It will contain the generated Q-LINK absolute program elements. This file is loaded when the COMUS INSTALL runstream is executed.
6	This file contains various Q-LINK request program examples and utilities. These examples may be examined, used as is or modified. This file is included for informational purposes only snd is loaded when the COMUS INSTALL runstream is executed.
7	This file is initially empty. It will be used for the Q-LINK generation print file, if the "print to file" option is selected in the COMUS BUILD.
8	This file, like the print file, is empty on the release tape but will be cumulative from generation to generation at your site. This file will contain the updated symbolics of any elements processed during BUILDs at your site. These symbolics can be used for input to CULL/IACULL if you wish such output.
9	For stability and update releases, this file contains a summary of changes since the previous base release.
10-11	These files are COMUS support files.

File numbers shown are relative to the beginning of the Q-LINK files on the release tape.

1.7 Quick Installation

The initial configuration contained on the distribution tape can be installed without requiring a Q-LINK generation. The Q-LINK parameter values used in the default generation can be found in Chapter 14, "Pre-GENed Q-LINK Parameters".

The initial configuration is made available by KMSYS Worldwide in order to provide our customers with a means of quickly installing and using Q-LINK without requiring a detailed understanding of the Q-LINK generation parameters. Once you become more familiar with the product, you may want to tailor the Q-LINK generation to the particular needs of your site.

This Quick Installation of Q-LINK may be used when installing InfoQuest for the purpose of evaluation.

The default installation can be accomplished using the COMUS INSTALL. The following procedures outline the steps necessary for quick installation with COMUS:

1. Register the release tape as shown in Section 3.2, "STEP-1: REGISTER Q-LINK".
2. Use the COMUS CONFIGURE to establish a configuration set as shown in Section 3.3, "STEP-2: CONFIGURE for BUILD/INSTALL".
3. Enter the product license information on the "Q-LINK Product License Information" screen.

> Be sure to enter the parameter values exactly as shown on the "Q-LINK Product License Key Information" sheet included with the release tape.

4. On the Q-LINK environment screen, "Environmental Parameters for Q-LINK Build/Install", set the Q-LINK IBANK NAMEs and BDIs to "NONE" (eight parameters).
5. On the Q-LINK installation screen, "INSTALL" (the continuation screen of the environment screen), change the bank name and BDI of Q-LINK's common data bank (CDB) if required and enter the qualified filename of the BIS relocatable output (MRO) file on your system. Note: The MRO file is only required for levels of BIS earlier than 35.

> The bank name and BDI must be unique for all products installed with AFCBs. You may determine if they have already been assigned by viewing SYS$*DATA$.CO$INSTALL$/COMUS$ with any text editor.

6. Exit the configuration process with the COMUS "E" command.
7. Follow the COMUS INSTALL procedure outlined in Section 3.5, "STEP-4: COMUS INSTALL". The MODE parameter shown on the INSTALL example is not required if using the default installation.
8. Use the COMUS CONFIGURE to issue the PROCESS command as described in Section 3.6, "STEP-5: Dynamic Runtime CONFIGURE".
9. Register the QUTIL run in BIS as discussed in Section 3.8 "STEP-7: Register QUTIL".
10. Verify the installation as shown in the Chapter 4, "Installation Verification Procedure".

The Q-LINK processor is installed as a program banked collection. If a common banked version of Q-LINK is desired, the COMUS BUILD and INSTALL procedures must be followed as illustrated in the remaining chapters of this installation guide.

Chapter 2: Installation Guidelines

The Installation Guidelines chapter will provide some definitions and explanations of terms used throughout the Installation Guide. Portions of this information will be presented again from an operating standpoint later in this manual. You may wish to read the operation chapter first to gain a better general understanding of the Q-LINK operating environment (see Chapter 7, "Q-LINK Operations").

2.1 The Q-LINK Common Data Bank (CDB)

Q-LINK servers communicate with BIS through a special shared memory area called the Q-LINK Common Data Bank (CDB). Only one CDB is normally required, even if multiple Q-LINK server configurations and multiple BIS systems will be used. One BIS system can communicate with multiple Q-LINK servers, but only through one CDB. Multiple BISs can use the same pool of Q-LINK servers through one Q-LINK CDB, or use totally separate Q-LINK servers through a separate Q-LINK CDB. If you have only one BIS, then you must have only one CDB in use (or, no more than one CDB per BIS system).

The only real need for multiple Q-LINK CDBs is when there is a site requirement to separate the interfaces physically between multiple BIS systems; however, the security feature of Q-LINK should be carefully examined before deciding that the installation of multiple Q-LINK CDBs is necessary (see Chapter 6, "Security").

Initial Q-LINK installation will include a processor called QMON. When you subsequently install a second Q-LINK that shares a previously installed CDB, the second installation does not include the QMON processor and is not separately configurable. Any configuration or administrative changes must be made by using the original installation's utilities (QMON execution, a COMUS CONFIGURE, etc.).

2.2 Maintaining Multiple Q-LINK Release Levels

Some sites may want to install Q-LINK to provide another execution level of Q-LINK; e.g., 6R5 can be used for production while 6R6 is being examined. If a second execution level is desired, a separate installation mode must be used so that both versions of

Q-LINK can coexist. In addition, the Common Data Bank (CDB) must be shared between the two installations.

If you are installing a second execution level of Q-LINK, please read Section 5.2, "Multiple Installations."

2.3 Common Banks vs. Program Banks

The Q-LINK Common Data Bank (CDB) must be installed as an Alternate File Common Bank (AFCB). The Q-LINK IBANKs may be configured as either AFCBs or program banks.

Configuring the Q-LINK processor to use common IBANKs is recommended, as common IBANKS can be shared by multiple Q-LINK users.

2.3.1 Server Common IBANKs

The Q-LINK processor(s) may be configured to use common IBANKs. Configuring the Q-LINK processor(s) to use common IBANKs is recommended, as common IBANKS can be shared by all servers (see page 2-4) in a like configuration.

If common IBANKs are used, you will need to select BDIs and bank names that are unique. Each version of the Q-LINK processor you build will use a different set of IBANKS.

2.3.2 Bank Naming and Bank Descriptor Index (BDI)

If you choose to change the name of the CDB, you must do so during the COMUS configuration process and prior to doing a COMUS INSTALL. The name given must be different for each Q-LINK common data bank used. If only one is to be used, the default name should be sufficient. In addition to the CDB, the Q-LINK processor's IBANKs may also be set up as common banks. The IBANK type and BDI assignment require a BUILD for the changes to take effect.

The BDIs for the Q-LINK banks must be entered in the configuration information if you have selected to use common banks. The BDIs must not be used by any other software currently installed on your system. The default BDIs were chosen from a range reserved for non-Unisys software and are not used by other KMSYS Worldwide software. To find out which BDIs are currently installed on your system, you may look at element:

```
SYS$*DATA$.CO$INSTALL$/COMUS$ .  EXEC 39 and higher
```

Unisys has allocated 816 out of a possible 4095 BDIs for site selection: indexes 0400300 through 0400323; starting with 0400400, the first 012 indexes of every 0100 (e.g., 0400700 through 0400711) up through 0403111; indexes 0403200 through 0403777; and indexes 0405500 through 0405777.

Bank names and BDIs are entered via the configuration screens for "Environmental Parameters for Q-LINK Generation" (see Section 3.3, "STEP-2: CONFIGURE for BUILD/INSTALL").

2.3.3 Start Address

The base address of the Q-LINK CDB determines where in the 262K user address space the Q-LINK CDB memory will be accessed. This address must be at least as high as the start of BIS's Multiple Memory Pool (MMP) banks, so as not to obscure the addressing of BIS's main memory pool. It must also be small enough so that the upper address does not overlap the main block of BIS's program control table (PCT) at 0777000. The size of the CDB can be approximately calculated as follows:

```
(number-of-server-classes * 24) + (maximum-active-servers * 20) + 10,000
```

Choosing a base address of octal 0777000 minus the CDB size and truncated to the next lower octal 01000-word boundary should be correct for most situations. The default CDB start address is 0700000 octal, which easily provides for a configuration consisting of a total of up to 10 available servers and up to 5 server classes. The default start address will work with all but very large BIS systems (refer to your BIS installation guide for BIS bank sizing).

2.4 Multiple DMS/RDMS Access

In many sites, more than one DMS 2200 and/or RDMS 2200 installation may be in use. A single version of the Q-LINK processor can link to a maximum of nine (9) different multi-

thread DMS/RDMS installations. Access to the multi-thread environment may be dynamically configured into Q-LINK without having to regenerate the product.

The linking to a particular DMS installation is accomplished through the standard COBLNK and LINKER relocatable routines supplied with the DMS or through a common-banked COBLNK relocatable element, CBEP$$DMS. For DMS installations under the Universal Data System (UDS), only the common-banked COBLNK option is allowed. Also, since UDS uses ALIAS processing when linking a schema to a particular application group, only the default application group need be configured in Q-LINK. For more information regarding multiple application groups, see Chapter 10, "MCB Considerations".

Linking to a particular RDMS installation is accomplished by specifying the EXEC application group name under which the RDMS application executes.

2.5 Q-LINK Servers – TIP vs. Batch

Q-LINK processes requests from BIS via Q-LINK processors, which execute as separate server programs. These server programs can be either batch runs or TIP transactions. When a Q-LINK server starts, it will link itself to the Q-LINK common data bank. At that time, it will be validated as a legal server class and placed into an idle/available state. In this idle/available state, the server program is deactivated by the EXEC and placed on a test-and-set queue. The server will be activated when a Q-LINK request program is routed to it from a BIS run through the CDB. This interaction is described more fully in the operating portion of this manual.

Before installing Q-LINK, you will need to make some choices as to how server programs will operate.

The same Q-LINK absolute can run as a batch program, a TIP transaction or a TIP on-line batch transaction. The runtime parameter, "BATCH, COMPOOL OR APPL NBR" only controls how Q-LINK auto-starts its servers. At any time, you can start any server manually (as a batch job), even if the server is setup as a TIP transaction (see Section 7.3.3, "Starting Q-LINK Servers").

Chapter 3: Installing Q-LINK

3.1 Using COMUS

COMUS is required to generate, install and configure Q-LINK. Please refer to your Series 2200 COMUS End Use Reference Manual, 7830 7758, for general instructions on using COMUS.

COMUS help screens are available for all Q-LINK parameters by typing a question mark (?) on any prompt.

In general, the COMUS method of Q-LINK installation will require the following steps:

1. REGISTER the Q-LINK release tape into the COMUS database.
2. Set up your local configuration parameters using the COMUS CONFIGURE command (do not issue a PROCESS command yet). The COMUS configure interface provides a simple "fill in" format to complete the configuration. On-line help is available for every configuration parameter.
3. Perform a product BUILD.
4. INSTALL Q-LINK.
5. Establish the runtime configuration using the COMUS CONFIGURE and PROCESS commands.

Many of the Q-LINK parameters can be changed without the necessity of performing a BUILD and INSTALL. By using the runtime configuration step (see Section 3.6, "STEP-5: Dynamic Runtime CONFIGURE"), these parameters can be changed as often as necessary without incurring unnecessary downtime. Runtime parameters that may be changed dynamically in this release are:

DMS 2200 access
Data name inclusion on INVOKE
DMS 2200 rollback upon exit
D$WORK NTRX for large PA/IPA loads
Significant characters when specifying area/record/set names
Multi-thread DMR access (up to 9 DMRs may be configured)
Common-banked or non-common-banked COBLNK file/elt/ver
LINKER file/elt/ver for non-common-banked COBLNKs
Default schema files
Default schema names
RDMS 2200 access
RDMS 2200 applications (up to 9 applications may be configured)
CBEP$$RSA file names;
RDMS/RSA PLUS stack size
RSA work space
RDMS 2200 rollback upon exit

- RDMS 2200 step advance if no END_THREAD
- RDMS 2200 runs with user's BIS ID
- BIS DTM access
- EXEC/FCSS/TIPDMS direct I/O
- User/account security lists for:
  - Q-LINK utilities
  - DMS 2200 schemas
  - DMS 2200 subschemas
  - DMS 2200 schema files
  - DMS 2200 DMRs
  - Default DMR
  - PCIOS files
  - PCIOS file types
  - RDMS 2200 applications
  - BIS DTM queues
  - DIO commands
  - DIO file types
- Privileged user password
- Configuration version
- Application group names (up to 9 may be configured)
- MCB CBEP$$MCB file names
- SFS 2200 CBEP$$SFS file names
- Maximum servers
- Record Delivery Area (RDA) size
- Default printer site
- Five alternate printer sites
- Alphabetic character range
- Default value for high-values
- Default source/object library qualifier
- Decimal point type (period or comma)
- DDP-PPC enable
- Class name
- Class-to-version association
- Maximum result lines
- Maximum request queue size
- Request queue warning level
- Maximum servers in a class
- Initial servers in a class
- BATCH/COMPOOL/MCB server execution
- PRODUCTION/DEVELOPMENT mode
- Enable/disable resource logging
- Maximum idle time
- Maximum SUP time
- Enable/disable automatic server start-up
- Temporary/cataloged server log file
- Diagnostic message level.

The dynamic runtime configuration parameters, listed above, are not used during the generation process.

All other Q-LINK parameters are entered using the COMUS CONFIGURE feature prior to the BUILD (see Section 3.3, "STEP-2, CONFIGURE for BUILD/INSTALL"). Once the BUILD process has been completed, they may not be changed again without performing another BUILD.

3.2 STEP-1: REGISTER Q-LINK

The first step in installing Q-LINK is to register the Q-LINK product release tape with your COMUS database. You may register Q-LINK with an existing COMUS database, or use COINIT to create a separate COMUS database for Q-LINK. The following is an example of a Q-LINK COMUS registration using a separate COMUS database that has already been initialized. User input is shown in boldface font.

```
▶@qual qkms .  Or your COMUS database qualifier
▶I:002333 QUAL complete.
▶@comus
▶ COMUS 6R8D (060816 1323:09) 2007 Apr 18 Wed 1321:58
▶Copyright (c) 1995-2006 Unisys Corporation.
▶COMMAND ? ▶register,s reel=660000 product=qlink,6r6
▶DATABASE DOES NOT CONTAIN A KEYWORD DICTIONARY.  NO KEYWORDS INSERTED.
»REGISTER PRODUCT 'QLINK 6R6' -<Y>/N/E(END) ?
▶
▶************ PRODUCT INFO FOR 'QLINK 6R6' ***************************
```

Because the "S" option was used on the REGISTER command, a series of screens containing current Q-LINK installation information will be displayed. You should read this information carefully. Note: This same information is contained on the first file (Q0) of the release tape in the element, "INFO/6R6" (the version name will change with each subsequent release level; e.g., 6R6A, 6R6B, etc.). Once you have browsed through the information pertaining to this release level, you should see the following:

```
▶******************** END INFO ************************************
▶HIT TRANSMIT TO CONTINUE OR T(TOP) »
▶@BRKPT PRINT$/022191095934 .
▶@SYM 022191095934.,,PR .
▶@CO$COS$PF.COMUS,ZT ,KMS
▶ TASK COMPLETED ************************
▶COMMAND ? »exit
▶END COMUS
```

The print file above shows the COMUS registration process, and is of little use to the user. The responses shown above may vary at your site due to differences in your COMUS default settings or COMUS release level.

> The COMUS BUILD and CONFIGURE facility provides a significant, ease-of-use method for building and configuring Q-LINK. If at your site, the primary COMUS database is unavailable to the person(s) maintaining Q-LINK, you may choose to set up an alternate COMUS database to be used for Q-LINK build and configuration activities. These functions will not generally require special privileges and will in no way interfere with the operations using the site's primary database.

3.3 STEP-2: CONFIGURE for BUILD/INSTALL

While portions of Q-LINK can be configured dynamically after its installation, several parameters can only be set prior to building the product (see Section 3.4, "STEP-3: COMUS BUILD", page 3-16). The COMUS CONFIGURE for Q-LINK is used to interactively create all BUILD parameters. The following pages show the configuration screens containing the parameters which must be set prior to performing a COMUS BUILD.

A detailed explanation of each parameter affecting the COMUS BUILD can be found by typing a question mark (?) over the first position of any given parameter and transmitting.

3.3.1 Entering the CONFIGURE Menus

To begin setting up your product configuration under COMUS you must first enter the COMUS CONFIGURE command. The following is an example of executing the CONFIGURE command:

```
▶@qual qkms
▶I:002333 QUAL complete.
▶@comus
▶COMUS 6R8D (060816 1323:09) 2007 Apr 18 Wed 1321:58
▶Copyright (c) 1995-2006 Unisys Corporation.
▶COMMAND ? ▶configure mode=update
```

At this point COMUS should go into full-screen mode. If your terminal is not in full-screen mode, type "\SCR" and transmit. This command will give you full-screen capability if your terminal type supports full-screen operations (UTS compatible terminals).

In full-screen mode, COMUS will display the first configuration menu. However, if no user set names are present in the COMUS database, it will be necessary to create one by using one of the COMUS COPY commands shown below.

For a new configuration set with KMSYS Worldwide release level defaults for all parameters, create a set for the new Q-LINK release level using the following command:

COPY PRODUCT=QLINK LEVEL=6R6 TO=*your-set-name*

If the set name (your-set-name) already exists in the COMUS database and contains a previous configuration of Q-LINK, the previously configured parameter values will be replaced by the release defaults; i.e., any site-specific values must be re-entered (see the COPY MERGE command below).

For a new configuration set that retains existing (i.e., site specified) parameter values, create a set for the new Q-LINK release level using the COPY MERGE command:

COPY MERGE FROM=*old-set-name* TO=*new-set-name* PRODUCT=QLINK LEVEL=6R6

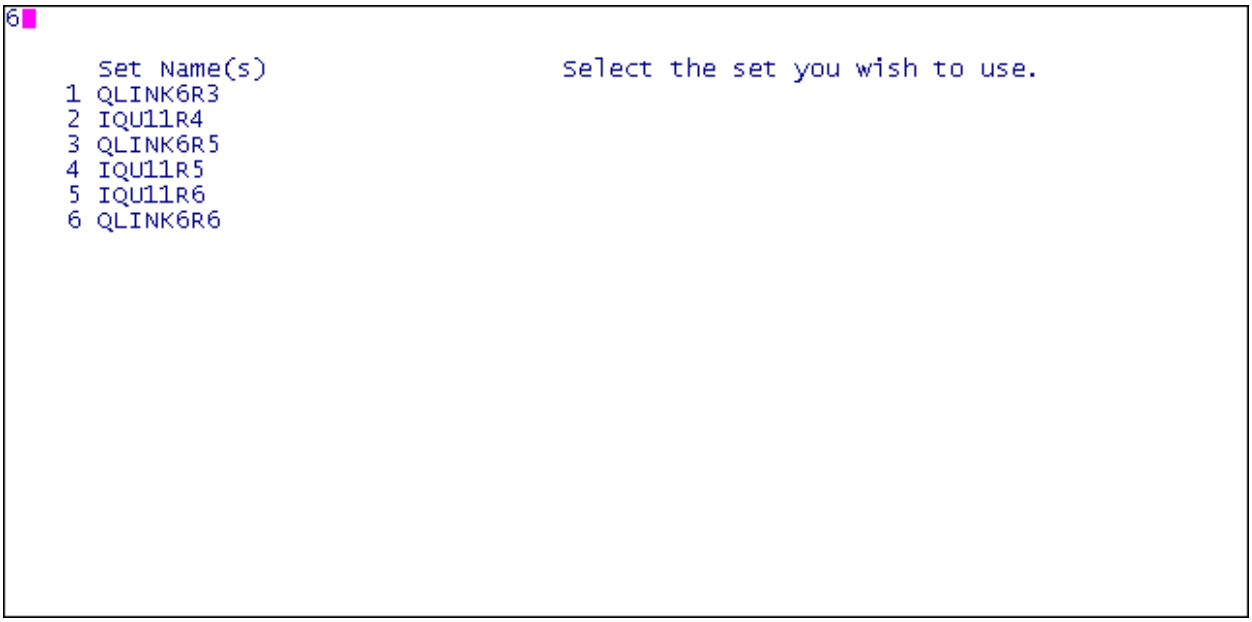
The COPY MERGE retains any settings of existing configuration parameters and establishes the default values for any new parameters. This command will allow you to migrate from one release level to another, without having to re-enter parameter values that were set for the previous release level.

The COPY MERGE command can also be used to provide another execution level of Q-LINK; e.g., 6R5 can be used for production while 6R6 is being examined. If a second execution level is desired, a separate installation mode must be used so that both versions of Q-LINK can coexist (see the MODE parameter illustrated in Section 3.5.2, "Performing the INSTALL").

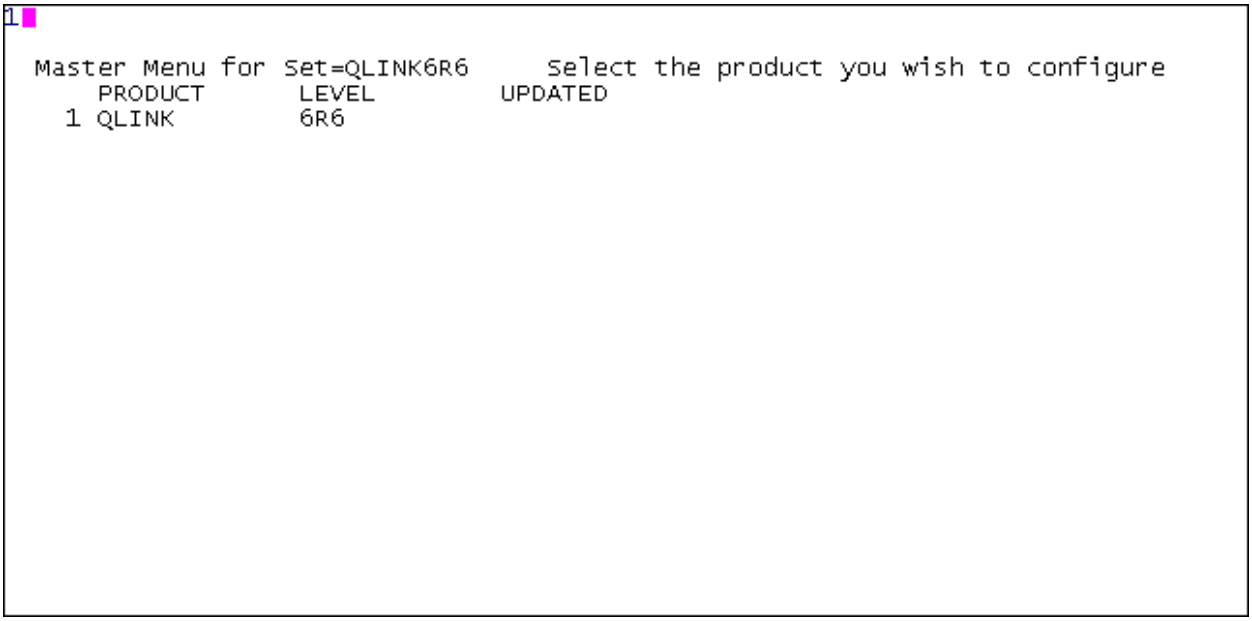
If you are installing a second execution level of Q-LINK, please read Section 5.2, "Multiple Installations".

When you are ready to put the new release level into production, you can use the COMUS COPY command to copy from the test configuration set to the production set and use the COMUS DELETE command to delete the test set. This procedure will require a reinstallation of Q-LINK utilizing the production installation mode. In addition, the product file name on the "Interface dynamic configuration parameters" screen should be changed back to the original product file name.

From the opening menu, select the user set you wish to configure. In this case, three Q-LINK configuration sets exist for three different release levels. The set named QLINK6R6 has been selected. Change any parameter values shown on the screens of this set and then use the COMUS BUILD command to generate a BUILD runstream. During the BUILD dialog, answer the configuration query with this configuration SET name and the BUILD will extract your parameter choices from the COMUS database and use them to tailor a Q-LINK for your site's particular needs.



Next, selecting product QLINK will cause the main Q-LINK configuration menu to be displayed.



3.3.2 Q-LINK Main Configuration Menu

Configuration processing through COMUS replaces the usage of generation SGSs for configuring the Q-LINK processor and utilities. While some Q-LINK parameters are still generation dependent (COMUS BUILD), ALL configuration information is entered via COMUS configure screens. The BUILD process will extract the generation time configuration information from the COMUS database in order to tailor table sizes and other BUILD dependent information. Many Q-LINK parameters are now dynamically configurable and can be changed by the system administrator without requiring a reBUILD of Q-LINK. The parameters in the default configuration set should provide a minimally functional Q-LINK. You can and should tailor this configuration to your site's requirements as you become familiar with the product.

```
36
 Enter + or - to view continuations of this screen.
                                                                SCREEN: MAIN
  This is the main menu for updating the configuration parameters for Q-LINK
  release 6R6 systems. You may use COMUS commands such as PROCESS, EXIT, OMIT,
  BACK, MODE=UPDATE, '?', '+' and '-' or enter a number to select from
  the following categories:

     1 .. 10) Server dynamic parameters for classes 1 .. 10   SCREEN: CLASSn
              Use '+' or '-' to access the menu screens for
              the server classes 11 .. 30 or enter
              DISPLAY SCREEN = CLASSn for a specific class.

     Product file ( SYS$LIB$*QLINK )  Version (  )
     31) Interface Dynamic Configuration Parameters         SCREEN: GLOBAL
     32) Basic Q-LINK Dynamic Configuration Parameters      SCREEN: BASIC
     33) Global Dynamic Parameters for DMS1100 Access       SCREEN: DMS1100
     34) DMR Names and Unique Information                   SCREEN: DMR1
     35) RSA/RDMS Application Names and Unique Information  SCREEN: RSA1

     36) Q-LINK Product License Information                 SCREEN: LICENSE
     37) Environmental Parameters for Q-LINK Build/Install  SCREEN: ENVIRON
     38) Basic Parameters for Q-LINK Generation             SCREEN: BUILD
```

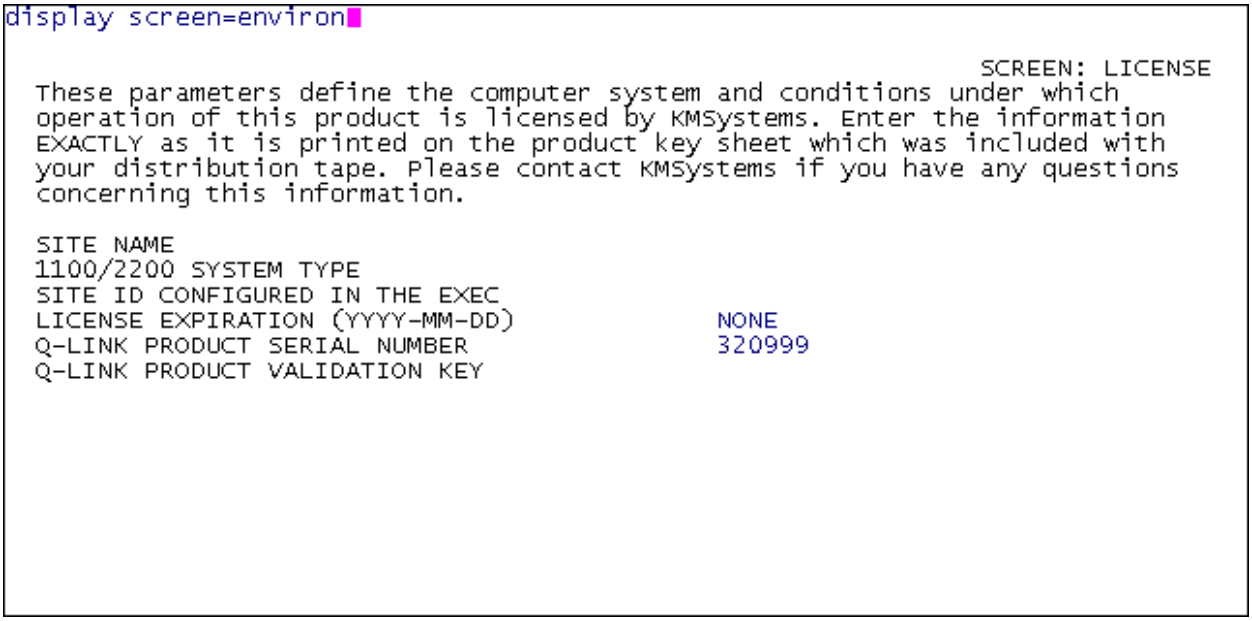
Normally, the first parameter values to enter are those required to validate operation of the software for the licensed computer system. Selection 36 will present the screen for "Q-LINK Product License Information".

Since a COMUS BUILD is required prior to processing the dynamic runtime configuration parameters (the COMUS CONFIGURE PROCESS command), the selections required for a BUILD (36 through 38) will be covered first in this guide.

A detailed explanation of each parameter affecting the COMUS BUILD can be found by typing a question mark (?) over the first position of a particular parameter and transmitting.

3.3.3 Q-LINK Product License Information

This information is used by Q-LINK to validate operation of the software for the licensed computer system. You have received a unique product key license sheet for each computer system for which you have been licensed for this software package. Please contact KMSYS Worldwide if the product key information you have received does not agree with the hardware configuration which you utilize for operation of this software package, or if you have any questions concerning the setup of this information. The entries in these fields MUST match the product key license information you received with the product unless you are advised otherwise by KMSYS Worldwide personnel.



The DISPLAY command can be used to access the next screen if the screen name is known, or use the BACK command to return to the main menu.

The values entered on the product license screen are required for BUILD/INSTALL and runtime processing. If they are not entered exactly as shown on the product key sheet packaged in the release shipment, validation errors will occur. These errors may occur at any stage (generation, installation or execution). If this occurs, first check the information entered on this screen against the product key sheet for correctness. If it appears to be correct, determine the system type and site-id of your system (as generated in the EXEC) by entering the following SSG runstream:

```
@SSG
SKEL
*DISPLAY 'SITE=([SYSTEM$,1,1,2]) SYSTEM TYPE = ([SYSTEM$,1,3,1])'
@EOF
@EOF
```

If you have taken the steps described above and are still unable to execute Q-LINK, please call KMSYS Worldwide for assistance.

If you need to BUILD Q-LINK on a system other than the one licensed for operation, please refer to Chapter 15, "Installing Q-LINK across Systems".

3.3.4 Environmental Parameters for Q-LINK Build/Install

The parameters on the following screen may only be changed with a Q-LINK generation (BUILD). Their values are used to supply the generation process with the location of the permanent SGSs and relocatable files. They also provide the optional processor names used for installation

Enter your choices for the parameter values shown on these screens and then use the COMUS BUILD command to create a generation runstream and the COMUS INSTALL command to create an installation runstream. By entering the name of the COMUS configuration set during the BUILD and INSTALL dialogs, COMUS will extract your parameter choices from the COMUS database and use them to tailor a Q-LINK for your site's particular needs.

```
+
Enter + or - to view continuations of this screen.
                                                          SCREEN: ENVIRON
 These parameters define information which can only be changed by a Q-LINK
 system generation. Enter a question mark (?) in any field and transmit for
 information on how to configure that parameter. Other generation parameters
 can be viewed with selections  from the main menu (use the BACK command).


 FILENAME OF MISC PARAMETERS ELT
 ELEMENT NAME OF MISC PARAMETERS
 OBJECT COMPUTER TYPE                          THIS SYSTEM
 USE COMMON BANKED SORT LIBRARY                THIS SYSTEM
 ACOB LIBRARY CBEP$$ACOB FILE                  THIS SYSTEM
 ACOB C$DML BANK TYPE                          THIS SYSTEM
 ACOB C$DML BANK SIZE                          THIS SYSTEM
 EXECUTION COMMON IBANK NAME                   QLKIBANK1
 EXECUTION COMMON IBANK BDI                    0400710
 EDITING COMMON IBANK NAME                     QLKIBANK2
 EDITING COMMON IBANK BDI                      0400711
 DMS I/F COMMON BANK NAME                      QLKIBANK3
 DMS I/F COMMON BANK BDI                       0400705
 FEATURE COMMON BANK NAME                      QLKIBANK4
 FEATURE COMMON BANK BDI                       0400706
```

Five of the generation parameters (those circled above) have a default value of "THIS SYSTEM". When set to "THIS SYSTEM", the generation process will automatically determine the operating environment for Q-LINK.

If you are migrating from an earlier release level and you used the COMUS COPY MERGE to retain the parameter settings of the earlier configuration set, you may set these parameters to "THIS SYSTEM" in order to simplify the generation process.

If you desire to use this automated feature, make sure that you do not have a permanent SGS that points to an ACOB library (e.g., INCLUDE or EXTRALIB) left over from a previous BUILD of Q-LINK. This permanent SGS may have been entered when the products defaults were defined during a previous COMUS BUILD session. You may view and/or reenter any permanent SGSs entered in this manner by using the "Q" option on the COMUS BUILD. Another place that permanent SGSs may reside is in the "MISC PARAMETERS" element, which may have been named on this same screen.

If you are building on one system to be installed on another system, you must ensure either that the configuration of these five parameters will not be different between the two systems, or you must explicitly fill in these parameters for the system on which Q-LINK will execute.

The parameters, "FILENAME OF MISC PARAMETER ELT" and the "ELEMENT NAME OF MISC PARAMETERS", are used to point to the element containing the permanent SGSs that may be required for Q-LINK generation (see Section 3.4.1, "Permanent SGSs", page 3-16). You must place these SGSs in this element prior to starting the Q-LINK generation runstream.

The environmental parameters on the following screen can be changed by a Q-LINK installation. Enter your choices for the parameter values on these screens and then use COMUS to generate an install runstream. Answer the configuration query with this configuration set name and the INSTALL will extract your parameter choices from the COMUS database and use them to install a Q-LINK for your site's particular needs. Changes to these parameters do not require a reBUILD unless you are changing the bank type.

```
disp scr=build
 Enter + or - to view continuations of this screen.
                                                          SCREEN: INSTALL
  These parameters define information which is used when INSTALLing Q-LINK.
  Changes to this information require a re-install of Q-LINK before they can
  take effect. Note: the CDB BDI must indicate an unused Alternate File
  Common Bank (AFCB) or Configured Common Bank (CCB) BDI. Enter a '?' in
  any field for detailed information.

  Q-LINK COMMON DATA BANK BDI                  0400707
  Q-LINK FIXED GATE SUB SYSTEM BDI
  Q-LINK COMMON DBANK NAME                     QCOMCB$D
  Q-LINK C-D-B START ADDRESS                   0740000
  CREATE SYS$LIB$*RUN$ RUNSTREAMS              TRUE
  MAPPER MASM PROCEDURE FILE (MRO)             NONE
  SUPUR FILE CODE FOR TXN SERVERS              0
  QLINK VALTAB NAME KEYWORD VALUE

  INSTALL NAME FOR QLINK PROGRAM
  INSTALL NAME FOR QINDEX PROG
  INSTALL NAME FOR QMON PROGRAM
  INSTALL NAME FOR QLSIM PROGRAM
  INSTALL NAME FOR QLKDDP PROGRAM
```

Setting the Q-LINK FIXED GATE SUB SYSTEM BDI has two effects:

1. Enables user-id matching;
2. Enables Common Data Bank protection for Security Option 3 sites.

BIS access to Q-LINK will not work when the Security Option Common Bank Protection feature is enabled.

Only sites requiring Security Option 3 should consider setting this parameter.

The setting of the fixed gate parameter above implies the following:

1. Since all transaction servers are started with the user-id of the executing QMON, transaction servers should not be configured unless all requestors are executing under the same EXEC user-id.

2. In the case of multiple user-ids requesting servers of the same class, and if the servers are allowed to remain idle for an extended period, it is possible to "lock out" a requestor even though there are idle servers present.

Example:

1. Maximum servers in the PROD1 class is three (3);
2. User-id A makes a request to PROD1, and PROD1 goes idle upon completion of the request;
3. User-id B makes a request to PROD1 and gets a second copy of PROD1 even though user-id A's copy of PROD1 is idle;
4. Likewise, User-id C makes a request to PROD1 and gets yet a third copy;
5. User-id D makes a request to PROD1 and is queued even though there are three idle copies in core. User-id D's request is queued because none of the idle servers pass the user-id selection criteria; however, if another request is generated by user-id B (or A or C), it will be processed immediately.

In this environment, it would be appropriate to limit idle time.

The "MAPPER MASM PROCEDURE FILE (MRO)" configuration parameter in COMUS, used during execution of the COMUS INSTALL runstream for Q-LINK, may be set to "NONE" for those sites installing Q-LINK to interface with BIS 35 or higher. This parameter is only required when interfacing with levels of BIS prior to 35.

3.3.5 Basic Parameters for Q-LINK Generation

These parameters must be changed with a Q-LINK generation (BUILD). Their values are used to determine the amount of table space required for Q-LINK.

```
e

                                                              SCREEN: BUILD
These parameters define information which can only be changed by a Q-LINK
system generation. Enter a question mark (?) in any field and transmit for
information on how to configure that parameter. DMS1100 related parameters
can be viewed with a selection from the main menu (use the BACK command).


MAX VARIABLE NAMES                          400
DATA STORAGE AREA (DOUBLE WORDS)            800
PROGRAM STORAGE SIZE (COMMANDS)             1400
MAXIMUM LABELS IN A PROGRAM                 400
MAXIMUM FILES (DEF F DIRECTIVES)            10

Q-LINK DBANK START ADDRESS                  0154000
DB4 PRODUCT LIBRARY FILE                    NONE
DTM PROC FILE
DTM SCHDLR INTERFACE BDI                    0403133
QWIZZ PRODUCT FILE                          NONE
DDP-PPC RELOCATABLE FILE                    NONE
DDP-PPC PROC FILE                           NONE
```

Once all of the parameters required for Q-LINK build have been entered, the COMUS "E" command can be used to terminate the COMUS configuration process in preparation for performing the COMUS BUILD. Note: the EXIT command will terminate the COMUS session.

3.4 STEP-3: COMUS BUILD

Once Q-LINK is registered with COMUS, you may proceed with the build process. This process will generate Q-LINK for installation in your environment. Before beginning the COMUS BUILD, you will need to create your local Q-LINK configuration SGSs that will be requested during the COMUS build. These SGSs are described below.

3.4.1 Permanent SGSs

The first group of SGSs defines the default processors and libraries to be used in the generation. These SGSs should be placed in the miscellaneous parameters element described earlier in this chapter (see Section 3.3.4, "Environmental Parameters for Q-LINK Build/Install"). Alternately, they could be supplied as permanent SGSs when COMUS queries for the permanent SGSs during the BUILD process. The defaults for the processor SGSs are:

```
CO$ACOBPROCESSOR CALL NAME IS ''ACOB'' ;
            OPTIONS ARE ''CES'' ;
            LIBRARY FILE IS ''SYS$LIB$*ACOB''
CO$EDPROCESSOR CALL NAME IS ''ED'' ;
            OPTIONS ARE ''N''
CO$ELTPROCESSOR CALL NAME IS ''ELT'' ;
            OPTIONS ARE ''LV''
CO$MAPPROCESSOR CALL NAME IS ''MAP'' ;
            OPTIONS ARE ''S''
CO$MASMPROCESSOR CALL NAME IS ''MASM'' ;
            OPTIONS ARE ''SEVY''
CO$PDPPROCESSOR CALL NAME IS ''PDP'' ;
            OPTIONS ARE ''L''
CO$PLSPROCESSOR CALL NAME IS ''PLS'' ;
            OPTIONS ARE ''SEF'' ;
            LIBRARY FILE IS ''SYS$LIB$*PLS''
CO$SSGPROCESSOR CALL NAME IS ''SSG''
CO$TCONPROCESSOR CALL NAME IS ''NOT''
CO$DCONPROCESSOR CALL NAME IS ''NOT''
CO$MCONPROCESSOR CALL NAME IS ''NOT''
CO$SORTPROCESSOR CALL NAME IS ''NONE'' ;
            OPTIONS ARE ''NONE'' ;
            LIBRARY FILE IS ''SYS$LIB$*SORT''
CO$SRTPROCESSOR CALL NAME IS ''NONE'' ;
            OPTIONS ARE ''NONE'' ;
            LIBRARY NAME IS ''SYS$*SRT$PAR''
CO$TIPPROCESSOR CALL NAME IS ;
            ''TIP$*TIPRUN$.SUPUR'' ;
            OPTIONS ARE ''NONE''
CO$PCIOSPROCESSOR CALL NAME IS ''NONE'' ;
            OPTIONS ARE ''NONE''
CO$SYSLIBPROCESSOR CALL NAME IS ''NONE'' ;
            OPTIONS ARE ''NONE'' ;
            LIBRARY FILE IS ;
            ''SYS$LIB$*SYSLIB''
CO$UCSRTSPROCESSOR CALL NAME IS ''NONE'' ;
            OPTIONS ARE ''NONE'' ;
            LIBRARY FILE IS ;
            ''SYS$LIB$*UCSRTS''
```

If these are not correct for your system, add permanent SGSs to your COMUS BUILD parameters to specify the correct files. Replace only those SGSs that must be different at your site.

Before continuing, please refer to Chapter 10, "MCB Considerations."

There are many different ways to install the ACOB processor and library at a given site. This can affect the way Q-LINK is installed. Please refer to Chapter 9, "ACOB Library Considerations", before continuing.

ACOB "extra options" can be specified by placing them in the second sub-field of the "OPTIONS ARE" clause, e.g., "OPTIONS ARE CES,NR".

If your site uses the ACOB "flagging" compiler (Unisys COBOL extensions are disabled), the "N" and "R" extra options must be specified in order to compile the Q-LINK programs.

The CO$SRTPROCESSOR is only used if you are not using common banked SORT libraries.

Q-LINK uses GSA for its input parsing and message generation. It is NOT necessary to have GSA installed for Q-LINK configuration and operation. If you have GSA level 5R1 installed and you must reprocess the GSA source modules used within Q-LINK, the CO$xCONPROCESSOR SGSs must define the location of the TCON, MCON and DCON processors. Only 5R1 may be used. An attempt to use any other level will cause sever errors. In addition, the "LIB FILE IS" clause on the CO$TCONPROCESSOR SGS must specify the file containing the updated source to the GSA library routines (normally GSA is installed in SYSLIB*GSA).

The BATCHPID SGS allows the PID number used by QMON for scheduling MCB based transaction servers to be changed (the default is PID 1). It only affects MCB servers (not COMPOOL). This SGS should only be used if PID 1 is being used for a real terminal or PID 1 is not configured in MCB. It should be noted that the BATCHPID is applicable to all application groups. Changing the BATCHPID can be accomplished with a MAPONLY generation. The *PID-number* shown below must be configured in MCB, but does not require any special configuration parameters:

BATCHPID *PID-number*

Before continuing, please refer to Chapter 10, "MCB Considerations".

If your libraries are installed in a non-standard manner and you require additional libraries to be searched and properly included in the Q-LINK collections, use the EXTRALIB SGS:

EXTRALIB || *qual-1*file-1* | *qual-2*file-2* | … *qual-n*file*n* ||

The INCLUDE SGS provides the means for including specific elements in the Q-LINK collection. This SGS is normally only used to include elements such as CERU$ (pre-COMUS libraries) or CBEP$$xxxx elements:

INCLUDE || *qual-1*file-1,elt-1* | *qual-2*file-2,elt-2* | … *qual-n*file-n,elt-n* ||

There may be only one INCLUDE SGS used in a Q-LINK generation. If multiple elements are to be included, you may continue this SGS on a new line by using the semicolon (;) continuation character as in the following example:

```
INCLUDE     QUAL*PF,ELT1 ;
QUAL*PF,ELT2
```

The NOT SGS allows the exclusion of elements which may be in your default library search path from the Q-LINK collections. As with the INCLUDE SGS, there may be only one NOT SGS used in a Q-LINK generation. If multiple elements are to be excluded, you may continue this SGS on a new line by using the semicolon (;) continuation character as illustrated in the previous example:

NOT || [*qual-1*file-1,*]*elt-1* | [*qual-2*file-2,*]*elt-2* | … [*qual-n*file-n,*]*elt-n* ||

If you wish to alter the listing and extra options fields of the CO$xyzPROCESSORSGSs, you may override both these fields by listing the processor type on a USEOPT SGS. Additionally, options for the @HDG statement may be specified here:

USEOPT || *processor-name-1,options-1,extra-options-1* | ;
processor-name-2,options-2,extra-options-2 | ;
… *processor-name-n,options-n,extra-options-n* ||

For example:

```
USEOPT ACOB,CES,K ELT,N MAP MASM,E PDP
```

3.4.2 Performing the BUILD

Once any additional SGSs have been set up, you may use the COMUS BUILD to generate the Q-LINK. The output of the build will be a product master tape ready to be installed. An example of a first-time Q-LINK COMUS build session follows:

```
▶@qual qkms
▶I:002333 QUAL complete.
▶@comus
▶COMUS 6R8D (060816 1323:09) 2007 Apr 18 Wed 1321:58
▶Copyright (c) 1995-2006 Unisys Corporation.
▶COMMAND ? ▶build qlink,6r6,
```

The following portion of the COMUS session will be used to set the product build defaults for the Q-LINK. You will only be prompted for the defaults the first time the product is built. On subsequent builds, COMUS will skip this section.

```
▶For each default question you may enter one of the following
▶responses:
▶ 1.  An appropriate value.
▶ 2.  A null string or spaces will maintain the current default value.
▶ 3.  QUERY - This keyword causes COMUS to ask for the default on
▶ every BUILD of the product.
▶ 4.  BLANK - This keyword sets the value of the default to null.
▶
▶Default project id (<Q$Q$Q$>) ? ▶qkms
▶Default run id (<>) ? ▶qlgen
▶Default run options (<>) ? ▶
▶Default run priority (<>) ? ▶d
▶Default tape equipment type (<T>) ? ▶
▶Default tape assign options (<TF>) ? ▶tj
▶Default generation type (DISK/DISK or <TAPE/TAPE>) ? ▶
▶Permanent SGSs (<END>) ? ▶
▶Printout: file, printer or query -F/P/<Q> ? ▶
▶Defaults Complete
```

This part of the COMUS session will occur on all builds.

```
▶Project id for this generation (<QKMS>) ? ▶
▶Run id for this generation (<QLGEN>) ? ▶
▶MASTER - reel/file./<> ? ▶660000
▶Generation id ? »qlink6R6
▶
```

```
▶Generation heading (<>) ? ▶Q-LINK Full Generation
▶
▶Generation reason (<END>) ? ▶New Generation of Q-LINK for Release 6R6
▶Generation reason (<END>) ? ▶
▶
▶New change number (<END>) ? ▶
▶
▶Perform ALL, FULL, CONFIG or MAPONLY gen -A/<F>/C/M ? ▶
▶NEWMASTER - reel/file./<> ? »newql
▶Printout to file or printer -<F>/P ? ▶
▶Additional SGSs (<END>) ? ▶
▶What is the QLINK configuration set name? ▶qlink6r6
```

At the above prompt, you should use the configuration set (named in STEP-2) containing your Q-LINK configuration parameters at the above prompt (see Section 3.3.1, "Entering the Configure Menus").

```
▶The runstream has been saved in 'QKMS*COMRUN(1).1/QLINK6R6'
▶View the runstream (Y or <N>) ? ▶
▶Print a copy of the runstream (Y or <N>) ? ▶
▶Start the runstream (Y or <N>) ? ▶
▶The runstream has been saved in 'QKMS*COMRUN(1).1/QLINK6R6'
▶UPDATING ACCESS FILES ...
▶ACCESS FILES HAVE BEEN UPDATED
▶BUILD TASK COMPLETED ************************
▶COMMAND ? »exit
▶END COMUS
```

If KMSYS Worldwide supplied changes have been inserted into the COMUS database (see Chapter 16, "Applying Changes to Q-LINK" in this guide), the change number(s) would be entered at the "New change number" prompt above. Normally, when you receive a release tape, there will be no changes to enter.

You must always perform a FULL or ALL generation type the first time the BUILD is run. The FULL or ALL generation will create the processors and BIS related elements. Subsequent CONFIG generations should then be used to configure the Q-LINK processor with various features. The MAPONLY generation can be used to change the banking or library structure defined when Q-LINK is mapped.

If a MASM level earlier than 5R1 is used for Q-LINK installation, a gen type of ALL must be done when the KMSYS Worldwide release tape is used as input to the gen.

The ALL forces the reprocessing of all elements, whether or not it is necessary, and should be used if you require full listings of all components generated.

The FULL generation causes all components of Q-LINK to be generated. This includes the Q-LINK processor, all Q-LINK common banks and QINDEX.

The MAPONLY type specifies that only the collection of the Q-LINK processors will be performed and the associated elements in the Q5 file will be generated. This may be done to change the names of AFCB IBANKs, or other portions of the Q-LINK processor MAPs. This type must not be used in your first Q-LINK generation.

It is normal to receive duplicate entry-point warning messages from the Collector when processing the Q-LINK maps for AFCB configurations; however, there should be zero ERRORs in all Q-LINK processor maps. The number of warnings in the main Q-LINK collection will equal the sum of the warnings in the IBANK-1, IBANK-2, IBANK-3 and IBANK-4 collections.

3.5 STEP-4: COMUS INSTALL

Once you have performed the Q-LINK product build, you can use COMUS to install the various components of the system. The install process will install the Q-LINK common banks, the executable Q-LINK absolutes and required Q-LINK runstreams. The following files and components will be installed:

3.5.1 The Product Files

SYSLIB*QLINK.	The file name will match the MODE parameter on the INSTALL command. The allowable modes are QLINK (the default), QLINKA, ... QLINKK. This file will contain all the executable processors, elements used to link Q-LINK to BIS, elements required for runtime configuration (see below) and all generated common banks (AFCBs) including common IBANKs for the Q-LINK processors and Q-LINK's common data bank.
SYSLIB*QLINK-1	This file (also dependent upon the MODE parameter on the INSTALL command) contains Q-LINK example request programs and utilities. It is provided for information and may be used for installation verification (see Chapter 4, "Installation Verification").

Processors found in SYSLIB*QLINK:

QLINK	Processor and, optionally, the four AFCB I-Banks.
QMON	Q-LINK operations interface.
QINDEX	Creates/updates/lists data item index files containing item definitions from schemas, subschemas and COBOL COPY proc formatted elements.
QLSIM	Used for InfoQuest batch processing and executing Q-LINK outside of BIS.
QLDDP	Used on the local system to process remote Q-LINK DDP requests. The QLDDP processor will only be generated if the required DDP-PPC files are provided when the Q-LINK software is generated.

The default names given to each of the five processors mentioned above are dependent upon the "INSTALL NAME FOR ..." configuration parameters (see Section 3.3.4, "Environmental Parameters for Q-LINK Build/Install") and the MODE parameter on the COMUS INSTALL command (review the COMUS SRL for the default processor names).

Other components found in SYSLIB*QLINK:

QLNK$CFIG	Q-LINK runtime configuration (built by the COMUS CONFIGURE PROCESS).
QUTIL	Q-LINK BIS utility run. This element must be retrieved into the BIS system after Q-LINK is installed into BIS (see Section 3.8, "STEP-7: Registering QUTIL").
CFIGSKEL	Element used in runtime configuration.
QLKCFIG	Element used in runtime configuration.
CO$CONFIG	Element that contains the generation and installation SGSs which produce this Q-LINK (used by CONFIGURE).
KMSLEV	Element contains the KMSYS Worldwide release level id and serial number for your copy of Q-LINK.
QLINK$DTM	BIS run that must be registered with BIS if the optional DTM interface is configured.

The INSTALL runstream must be run (@START) under either the site security officer's user-id or a user-id which has been granted the necessary privileges for installation. For further information, see the COMUS End Use Reference Manual, 7830 7758.

If installing multiple versions of Q-LINK, you must use the MODE parameter on the second and subsequent INSTALL command specifying a mode name which is not currently installed at your site; otherwise, the first install will be overlaid. The mode name chosen will be used as part of the Q-LINK product file names; e.g., if MODE=QLINKB, product files SYSLIB*QLINKB and SYSLIB*QLINKB-1 would be produced. In addition, the mode is used to generate the installed program names when not specifically named in the configuration parameters. If MODE=QLINKB, the following five processors would be produced: QLINKB, QMONB, QINDXB, QLBSIM and QLBDDP

Please view the COMUS System Registration Log (SRL) before using the MODE parameter.

Use of the INSFILE parameter on installations, whose CDB may be shared by a subsequent installation, is not allowed.

If this installation is to share a previously installed CDB, you must manually copy the QLNK$CFIG absolute from the shared product file to the new product file once the configuration has been processed by the COMUS PROCESS command (see Section 5.2, "Multiple Installations").

3.5.2 Performing the INSTALL

An example of the COMUS install session follows:

```
▶@qual qkms
▶I:002333 QUAL complete.
▶@comus
▶COMUS 6R8D (060816 1323:09) 2007 Apr 18 Wed 1321:58
▶Copyright (c) 1995-2006 Unisys Corporation.
▶COMMAND ? ▶install product=qlink,6r6 reel=newql mode=qlinkc
```

Note: If you receive a stability release of Q-LINK, enter the complete stability level (6R6A, 6R6B, etc.) on the COMUS INSTALL.

In the above example, an alternate installation mode (QLINKC) was used. An alternate mode should be used to avoid overlaying an existing installation of Q-LINK; e.g., a previous release level to be maintained until the new level is placed into production. Also, when a second installation mode is used in this manner, be sure that the BDIs, bank names and installed program names are unique.

The following portion of the COMUS session will be used to set the product install defaults for Q-LINK. You will only be prompted for the defaults the first time any product is installed. On subsequent installs, COMUS will skip this section.

```
▶For each default question you may enter one of the following
▶responses:
▶ 1.  An appropriate value.
▶ 2.  A null string or spaces will maintain the current default value.
▶ 3.  QUERY - This keyword causes COMUS to ask for the default on
▶ every INSTALL of the product.
▶ 4.  BLANK - This keyword sets the value of the default to null.
▶
▶Enter the default tape equipment type - <T> ▶
▶Enter the default tape assign options - <TJ> ▶
▶Enter the default project id - <> ▶qkms
▶Enter the default runid - <> ▶ql6ins
```

```
▶Enter the default run options - <> ▶
▶Enter the default run priority - <> ▶a
▶Enter any permanent SGS's for product QLINK 6R6
▶SGS or <END> ▶
```

This part of the COMUS session will occur on all INSTALLs. Remember, you can get more information on any prompt by simply transmitting a question mark (?).

```
▶Enter the project id for this installation - <QKMS> ▶
▶Enter reason for installation of QLINK 6R6
▶Reason or <END> ▶Q-LINK 6R6 Installation
▶Reason or <END> ▶
▶Enter any additional SGS's for QLINK 6R6
▶SGS or <END>▶
▶Will QLINKC share an existing QLINK CDB - Y/<N> ? ▶
```

If you are installing a second copy of Q-LINK that will share the same Common Data Bank (CDB), answer yes (Y) to the above prompt. Some sites may want to use this facility to provide another execution level of Q-LINK; e.g., 6R5 can be used for production while 6R6 is being examined. If a second execution level is desired, a separate installation mode must be used so that both versions of Q-LINK can coexist.

If you are installing a second execution level of Q-LINK, please read Section 5.2, "Multiple Installations".

```
▶What is the Q-LINK configuration set name? ▶qlink6R6
▶The runstream has been saved in 'QKMS*COMRUN(1).INS-16'
▶View the runstream - Y/<N> ? ▶
▶Print a copy of the runstream - Y/<N> ? ▶
▶Start the runstream - <Y>/N ? ▶n
▶The runstream has been saved in 'QKMS*COMRUN(1).INS-16'
▶INSTALL TASK COMPLETED ************************
▶COMMAND ? ▶exit
▶END COMUS
```

3.6 STEP-5: Dynamic Runtime CONFIGURE

In this step, you will be configuring the runtime attributes of Q-LINK. You may use this process whenever you wish to change Q-LINK's runtime parameters. Under the COMUS configuration described here, most runtime configuration changes will take effect immediately.

Before proceeding with configuration for the first time, it would be advisable to read the chapter on Q-LINK operation (see Chapter 7, "Q-LINK Operations") so that you will fully understand the effect of the various configuration options.

To perform dynamic configuration, enter COMUS and use the CONFIGURE command as in STEP-2. Adjust the dynamic configuration parameters as necessary, then enter the PROCESS command from any screen within the configuration set. COMUS will then go into scrolling screen mode and perform the configure routines.

The COMUS PROCESS command cannot be issued until the Q-LINK product file has been installed by the installation runstream (see Section 3.5, "STEP-4: COMUS INSTALL"). In addition, the installation file must not be assigned to any user and all servers and QMON executions must be terminated prior to issuing the PROCESS command.

The following is an example of executing the COMUS CONFIGURE command.

```
▶@qual qkms
▶I:002333 QUAL complete.
▶@comus
▶COMUS 6R8D (060816 1323:09) 2007 Apr 18 Wed 1321:58
```

```
▶Copyright (c) 1995-2006 Unisys Corporation.
▶COMMAND ? ▶config set=qlink6r6 product=qlink level=6r6 mode=update
```

Since the set and product names were supplied on the CONFIGURE command, COMUS will bypass the set and product selection screens.

The following pages will show each screen for the dynamic runtime configuration parameters. A detailed explanation of each parameter can be found by typing a question mark (?) over the first position of a particular parameter and transmitting.

```
+
 Enter + or - to view continuations of this screen.
                                                                SCREEN: MAIN
  This is the main menu for updating the configuration parameters for Q-LINK
  release 6R6 systems. You may use COMUS commands such as PROCESS, EXIT, OMIT,
  BACK, MODE=UPDATE, '?', '+' and '-' or enter a number to select from
  the following categories:

     1 .. 10) Server dynamic parameters for classes 1 .. 10   SCREEN: CLASSn
              Use '+' or '-' to access the menu screens for
              the server classes 11 .. 30 or enter
              DISPLAY SCREEN = CLASSn for a specific class.

     Product file ( SYS$LIB$*QLINK )  Version (  )
     31) Interface Dynamic Configuration Parameters          SCREEN: GLOBAL
     32) Basic Q-LINK Dynamic Configuration Parameters       SCREEN: BASIC
     33) Global Dynamic Parameters for DMS1100 Access        SCREEN: DMS1100
     34) DMR Names and Unique Information                    SCREEN: DMR1
     35) RSA/RDMS Application Names and Unique Information   SCREEN: RSA1

     36) Q-LINK Product License Information                  SCREEN: LICENSE
     37) Environmental Parameters for Q-LINK Build/Install   SCREEN: ENVIRON
     38) Basic Parameters for Q-LINK Generation              SCREEN: BUILD
```

Server classes 1, 2 and 3 are initially set up as PROD1, TEST1 and OFFLIN, respectively (see next page). The parameter values for PROD1 and TEST1 are initially set for batch on-line processing. The TEST1 server class is a diagnostic server (development) and is normally only used for debugging Q-LINK programs. OFFLIN is used for off-line batch reporting and has its default values set accordingly. Server classes 4 through 30 are not defined (named) in the default configuration.

As server classes are configured, each class name will appear on the continuation screen for the main configuration menu as shown below. PROD1, TEST1 and OFFLIN are initially configured at KMSYS Worldwide and are the default classes for this release.

```
+
 Enter + or - to view continuations of this screen.
                                                              TITLE: MAINEXT1
  This is the main menu for updating the dynamic parameters for Q-LINK
  release 6R6 servers. You may view or modify the runtime configuration
  information for the server class parameters and the global parameters.
  You may use COMUS commands such as PROCESS, EXIT, OMIT, BACK, MODE=UPDATE,
  and '?',  '+', '-' or enter a number to select from the menu:

     1)  Server Class ( PROD1 ) Parameters
     2)  Server Class ( TEST1 ) Parameters
     3)  Server Class ( OFFLIN ) Parameters
     4)  Server Class (  ) Parameters
     5)  Server Class (  ) Parameters
     6)  Server Class (  ) Parameters
     7)  Server Class (  ) Parameters
     8)  Server Class (  ) Parameters
     9)  Server Class (  ) Parameters
     10)  Server Class (  ) Parameters

     Product file ( SYS$LIB$*QLINK )  Version (  )
     31) Interface Dynamic Configuration Parameters          SCREEN: GLOBAL
```

If more than 10 server classes are required, they will be listed on two additional continuation screens. Up to 30 classes may be configured.

```
+
 Enter + or - to view continuations of this screen.

                                                              TITLE: MAINEXT2
  This screen displays additional classes for server runtime configuration.
  You may enter the index of the server whose configuration you wish to
  modify or you use may use COMUS commands such as PROCESS, EXIT, OMIT, BACK,
  MODE=UPDATE, '?',  or enter '+' or '-' to view continuation screens.

     11)  Server Class (  ) Parameters
     12)  Server Class (  ) Parameters
     13)  Server Class (  ) Parameters
     14)  Server Class (  ) Parameters
     15)  Server Class (  ) Parameters
     16)  Server Class (  ) Parameters
     17)  Server Class (  ) Parameters
     18)  Server Class (  ) Parameters
     19)  Server Class (  ) Parameters
     20)  Server Class (  ) Parameters

     Product file ( SYS
     31) Interface Dyna
```

```
display screen=class1
 Enter + or - to view continuations of this screen.

                                                              TITLE: MAINEXT3
  This screen displays additional classes for server runtime configuration.
  You may enter the index of the server whose configuration you wish to
  modify or you use may use COMUS commands such as PROCESS, EXIT, OMIT, BACK,
  MODE=UPDATE, '?',  or enter '+' or '-' to view continuation screens.

     21)  Server Class (  ) Parameters
     22)  Server Class (  ) Parameters
     23)  Server Class (  ) Parameters
     24)  Server Class (  ) Parameters
     25)  Server Class (  ) Parameters
     26)  Server Class (  ) Parameters
     27)  Server Class (  ) Parameters
     28)  Server Class (  ) Parameters
     29)  Server Class (  ) Parameters
     30)  Server Class (  ) Parameters

     Product file ( SYS$LIB$*QLINK )  Version (  )
     31) Interface Dynamic Configuration Parameters          SCREEN: GLOBAL
```

3.6.1 Server Dynamic Parameters for Classes

The following parameters can be changed without requiring a Q-LINK generation or installation.

For each class of server which you wish to create (e.g., test, production) you must define a class name and the associated runtime parameters which determine how the server class operates. The examples below are for Server Class 1.

```
+
 Enter + or - to view continuations of this screen.
                                                          TITLE: CLASS1
  These parameters define information which applies to a single server class.
  Enter a question mark (?) in any field and transmit for information on
  how to configure that parameter.



  CLASS NAME                                  PROD1
  CONFIGURATION VERSION (OPTIONAL)
  MAX RESULT LINES                            1000
  MAX REQUEST QUEUE SIZE                      5
  REQUEST QUEUE WARNING LEVEL                 3
  MAX SERVERS IN CLASS                        5
  INITIAL SERVERS IN CLASS                    0
  COMPOOL, BATCH OR APPL GROUP NBR            BATCH
  PRODUCTION OR DEVELOPMENT SERVER            PRODUCTION
  REQUEST RESOURCE LOGGING                    OFF
  MAX IDLE TIME (MINUTES)                     0
  MAX REQUEST SUP TIME (SECONDS)              262143
  SERVER RUNSTREAM/TIP ACTION CODE
  RUNSTREAM NAME (MORE)
  Q-LINK LIBRARY QUALIFIER                    Q$LNK
```

If CLASS NAME is changed, A QMON INIT must be performed before the new CLASS NAME takes effect.

The server type field that reads, “BATCH OR TIP EXECUTION”, on the server class screen must contain the words, “COMPOOL” or “BATCH”, or the number of the EXEC application group.

The following screen is the continuation screen for a server class:

```
disp scr=global
 Enter + or - to view continuations of this screen.
                                                          TITLE: CLASSEXT1
  These parameters define information which applies to a single server class.
  Enter a question mark (?) in any field and transmit for information on
  how to configure that parameter.


  PRIVILEGED REQUEST PASSWORD                 QLINKP
  USE TEMPORARY SERVER LOG FILE               TRUE
  DIAGNOSTIC MESSAGE LEVEL                    0
```

3.6.2 Interface Dynamic Configuration Parameters

The following parameters can be changed without requiring a Q-LINK generation or installation.

The MAXSERVERS parameter determines the maximum number of concurrent server runs which may be registered with the Common Data Bank (CDB) interface. A number in the range of 3 to 5 is probably sufficient for many sites. Each potentially active server slightly increases the size of the CDB.

When you tell COMUS to PROCESS the configuration set, it will write the configuration to the specified product installation file (as specified on this screen). Subsequent Q-LINK executions, once the common data bank (CDB) has been reloaded with a QMON INIT, will utilize the new parameter values. If the MODE parameter was used on the COMUS INSTALL, the INSTALLATION FILE name must be changed to match the mode used before doing the PROCESS.

```
disp scr=basic
 Enter + or - to view continuations of this screen.
                                                          SCREEN: GLOBALEX1
  These parameters associate related pieces of the Integrated Recovery
  environment.  Each continuation screen is associated with the specific
  application group number used in the group name parameter below.  These
  parameters can be changed without requiring a system generation (BUILD)
  of Q-LINK.  Enter a question mark (?) in any field and transmit for
  information on how to configure that parameter.

  Name of Application Group 3
  Name of MCB CBEP$$MCB3 file
  Name of SFS CBEP$$SFS3 file
  Name of Application Group 4
  Name of MCB CBEP$$MCB4 file
  Name of SFS CBEP$$SFS4 file
  Name of Application Group 5
  Name of MCB CBEP$$MCB5 file
  Name of SFS CBEP$$SFS5 file
  Name of Application Group 6
  Name of MCB CBEP$$MCB6 file
  Name of SFS CBEP$$SFS6 file
```

```
+
 Enter + or - to view continuations of this screen.
                                                             TITLE: GLOBAL
  These parameters define information which applies to all server classes
  regardless of the config version name.  The application group parameters
  associate related pieces of the Integrated Recovery environment.  These
  parameters can be changed without requiring a system generation (BUILD) of
  Q-LINK.  Enter a question mark (?) in any field and transmit for
  information on how to configure that parameter.

  Installation File                          SYS$LIB$*QLINK
  Configuration Version
  Maximum Concurrent Servers                 5


  Name of Application Group 1
  Name of MCB CBEP$$MCB1 file
  Name of SFS CBEP$$SFS1 file
  Name of Application Group 2
  Name of MCB CBEP$$MCB2 file
  Name of SFS CBEP$$SFS2 file
```

These GLOBAL screen and its extensions, GLOBALEX1 and GLOBALEX2, are where Step Control Application Groups, MCB and SFS 2200 entry points are configured.

3.6.3 Basic Q-LINK Dynamic Configuration Parameters

The following parameters can be changed without requiring a Q-LINK generation or installation. The values are used to turn ON/OFF interface access code, link to the security access element, establish printer queues, etc.

```
+
 Enter + or - to view continuations of this screen.
                                                              SCREEN: BASIC
  These parameters define information which can be changed without requiring
  a system generation (BUILD) of Q-LINK. Enter a question mark (?) in any
  field and transmit for information on how to configure that parameter.
  You can display other screens in this sequence by transmitting + or - at
  the home position.

  DB4 DATABASE ACCESS                          OFF
  QWIZZ INTERFACE                              OFF
  EXEC/FCSS/TIPDMS DIRECT I/O                  ON
  RDMS 1100 INTERFACE                          OFF
  DTM INTERFACE                                OFF
  DEFAULT PRINTER DEVICE/QUEUE                 PR
  ALT PRINTER DEVICE/QUEUE #1                  PR
  ALT PRINTER DEVICE/QUEUE #2                  PR
  ALT PRINTER DEVICE/QUEUE #3                  PR
  ALT PRINTER DEVICE/QUEUE #4                  PR
  SECURITY BY USERID/NONE                      NONE
  SECURITY ACCESS LIST SOURCE
```

If the SECURITY BY USERID/NONE parameter is set to "USERID", all access features (utilities, PCIOS files, DMRs, etc.) required by a site must be configured or access will be denied. Chapter 6, "Security" of this guide explains how to configure the security SGSs to be placed in the element and file specified on the SECURITY ACCESS LIST SOURCE parameter.

If a change is made to the SECURITY ACCESS LIST SOURCE, the Q-LINK common data bank (CDB) must be reloaded with the QMON INIT command for the new security to take effect. Execute the QMON INIT command after the COMUS CONFIGURE PROCESS.

It is possible to lock yourself out of Q-LINK by having only invalid or unusable user-ids in the security SGS list. If this happens you will not have access to QMON and consequently the Q-LINK common data bank (see Section 7.4.3, "Reinitializing the CDB Security")

The following is the continuation screen for the basic configuration parameters.

```
disp scr=dms1100
 Enter + or - to view continuations of this screen.
                                                          SCREEN: BASIC1
  These parameters define information which can be changed without requiring
  a system generation (BUILD) of Q-LINK. Enter a question mark (?) in any
  field and transmit for information on how to configure that parameter.
  You can display other screens in this sequence by transmitting + or - at the
  home position.

  RECORD DELIVERY AREA (WORD SIZE)            4000

  STANDARD/ALT HIVALS (0177/0377)             0177
  SPECIAL CHARS TREATED AS ALPHA
  EUROPEAN DECIMAL FORMAT                     FALSE
  DDP-PPC ENABLE FLAG                         OFF
```

The characters included in $ALPHA are selected with a bit map containing all alpha characters. The Q-LINK administrator can add/delete up to 32 special alpha characters. The default $ALPHA definition now includes A..Z (FIELDATA or ASCII), SPACE and a..z (ASCII only).

3.6.4 Global Dynamic Parameters for DMS 2200 Access

The following parameters can be changed without requiring a Q-LINK generation or installation. Their values are used to turn ON/OFF the DMS 2200 interface access code and establish features common to all DMRs/UDS Controls configured.

```
disp scr=dmr1

                                                          SCREEN: DMS1100
  These parameters define DMS1100 related configuration information which can
  be changed without requiring a system generation (BUILD) of Q-LINK. Enter a
  question mark (?) in any field and transmit for information on how to
  configure that parameter.


  DMS1100 ACCESS                              ON
  D$WORK NTRX FOR PA/IPA LOAD                 0
  INVOKE PROCESSES DATA NAMES                 TRUE
  ROLLBACK IF EXIT W/O DEPART                 FALSE
  SIGNIFICANT CHARS: AREA NAMES               12
  SIGNIFICANT CHARS: RECORD NAMES             30
  SIGNIFICANT CHARS: SET NAMES                30
```

3.6.5 DMR Names and Unique Information

The following parameters can be changed without requiring a Q-LINK generation or installation. The parameters on the following screens (only two out of nine possible are shown) are utilized to identify the DMRs/UDS Controls that can be configured for Q-LINK access.

```
+
 Enter + or - to view continuations of this screen.
                                                                  SCREEN: DMR1
  These parameters configure Q-LINK for access to a multi-thread DMS1100
  application as defined by a specific DMR. These parameters can be changed
  without requiring a system generation (BUILD) of Q-LINK. Enter a question
  mark (?) in any field and transmit for information on how to configure
  that parameter. Global parameters can be accessed from the main product menu
  (screen DMS1100).
  DMR1 MULTI-THREAD INVOKE NAME                MT
  DMR1 COBLNK/CBEP$$DMS FILE
  DMR1 COBLNK/CBEP$$DMS ELT/VER
  DMR1 RELOCATABLE LINKER FILE
  DMR1 LINKER ELEMENT/VERSION
  DMR1 BUFFER BANK BASE ADDRESS                0400000
  DMR1 DEFAULT SCHEMA FILE
  DMR1 DEFAULT SCHEMA NAME
```

The first multi-thread configured (SCREEN: DMR1) will be the default DMR used if the "FOR DMR-name" is not specified on the INVOKE directive. The security feature can override the defaults.

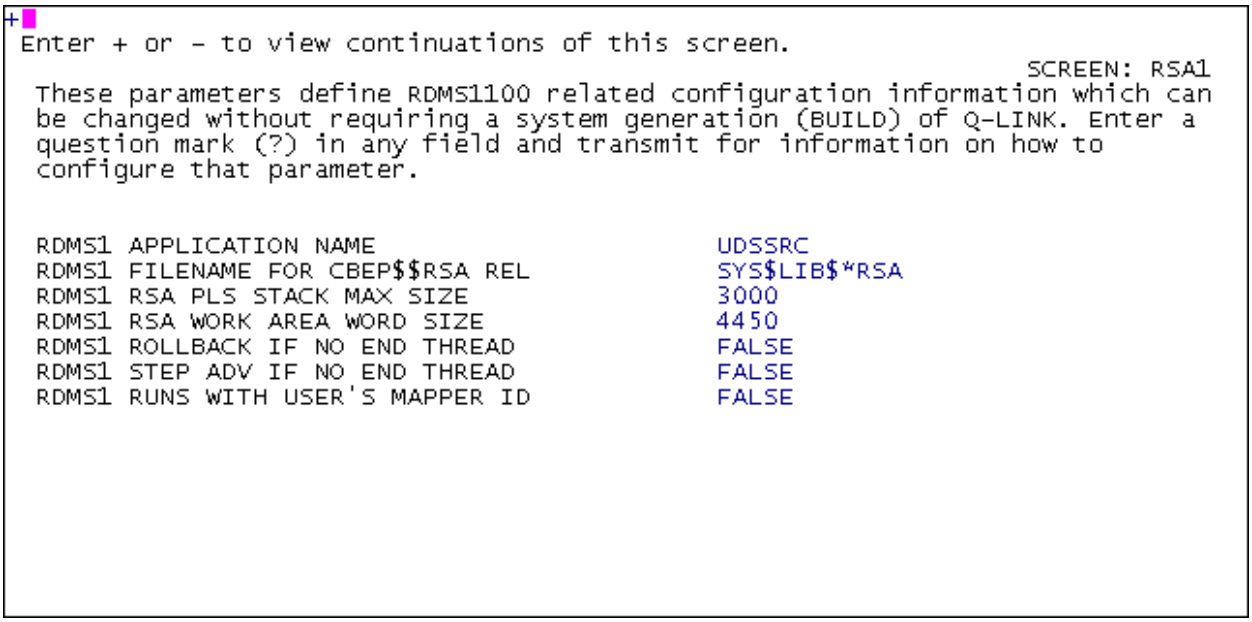
When configuring for DMS (level 9 or higher) under UDS Control, the "BUFFER BANK BASE ADDRESS" parameter may be configured to start at "0600000".

The following screen is the continuation screen for DMR1. This screen (DMR2) and the other continuation screens (DMR3 through DMR9) allow additional DMRs to be configured. This is normally not required for DMS level 9 and higher levels if schema ALIAS processing is used.

```
disp scr=rsa1
 Enter + or - to view continuations of this screen.
                                                                  SCREEN: DMR2
  These parameters configure Q-LINK for access to a multi-thread DMS1100
  application as defined by a specific DMR. These parameters can be changed
  without requiring a system generation (BUILD) of Q-LINK. Enter a question
  mark (?) in any field and transmit for information on how to configure
  that parameter. Global parameters can be accessed from the main product menu
  (screen DMS1100).
  DMR2 MULTI-THREAD INVOKE NAME
  DMR2 COBLNK/CBEP$$DMS FILE
  DMR2 COBLNK/CBEP$$DMS ELT/VER
  DMR2 RELOCATABLE LINKER FILE
  DMR2 LINKER ELEMENT/VERSION
  DMR2 BUFFER BANK BASE ADDRESS                0400000
  DMR2 DEFAULT SCHEMA FILE
  DMR2 DEFAULT SCHEMA NAME
```

3.6.6 RSA/RDMS Application Names and Unique Information

The following parameters can be changed without requiring a Q-LINK generation or installation. The parameters on the following screens (only two out of nine possible are shown) are utilized to identify the various RDMS application interfaces that can be configured for Q-LINK access.



Q-LINK must parse the BEGIN THREAD and END THREAD commands in order to track the status of thread control for the RDMS interface. It verifies the thread-name (APPLICATION NAME) is valid as configured in Q-LINK (this is also the basis for the security enforcement).

Up to nine RDMS applications can be configured in a single Q-LINK. The configuration screen for an RDMS application requires the filename containing the CBEP$$RSA element, choices for ROLLBACK and STEP handling and the stack and workspace sizes for that RDMS level. The RDMS-ACOBDAT interface module (ACOB interface to RSA) is NOT used.

```
process
 Enter + or - to view continuations of this screen.
                                                                SCREEN: RSA2
  These parameters define RDMS1100 related configuration information which can
  be changed without requiring a system generation (BUILD) of Q-LINK. Enter a
  question mark (?) in any field and transmit for information on how to
  configure that parameter.


  RDMS2 APPLICATION NAME
  RDMS2 FILENAME FOR CBEP$$RSA REL
  RDMS2 RSA PLS STACK MAX SIZE                3000
  RDMS2 RSA WORK AREA WORD SIZE               4450
  RDMS2 ROLLBACK IF NO END THREAD             FALSE
  RDMS2 STEP ADV IF NO END THREAD             FALSE
  RDMS2 RUNS WITH USER'S MAPPER ID            FALSE
```

The data space for the RDMS interface (stack, workspace and parameter storage) is dynamically allocated at runtime when a BEGIN THREAD is issued. The data is allocated in a special D-bank, and hence, it does not affect the size or addressing of the main D-bank for Q-LINK. This D-bank is allocated beyond the end of the Q-LINK process I-banks and before the start of the main D-bank, overlaying the address space of the utility I-bank. Thus, lowering the start address of the Q-LINK D-bank could restrict the size of the stack and workspace areas for RDMS. If this becomes a problem, sufficient space can probably be obtained by NOT using the C$DML common bank. This will add 022000 (or 037000) to the available area for the RDMS interface to acquire. In addition, it will reduce the size of the main D-bank by about 05400 words that may allow the start address of that bank to be increased, and thus, provide more room for the RDMS interface to acquire. The space dynamically acquired for the RDMS interface to use is released when the INIT command is performed.

3.7 STEP-6: BIS Start Parameter

If you are using BIS level 35, make sure that the BIS start parameter, QLKCDB, has the same CDB BDI (0400707 is the default) as was configured in COMUS for Q-LINK generation.

3.8 STEP-7: Registering QUTIL

After Q-LINK and the updated BIS system have been installed, Q-LINK is operational; however, KMSYS Worldwide provides a BIS run called QUTIL which will be of general use to all Q-LINK application developers. This run is referred to in the Q-LINK Applications Development Users Guide. QUTIL will be of use both in learning to use the capabilities of Q-LINK, and in providing the experienced Q-LINK user with a general-purpose run to submit requests to Q-LINK. The QUTIL run is in file SYSLIB*QLINK. The QUTIL run should be

placed in a form type generated with 132-character lines and the full character set. The BIS format option must not be used when retrieving the QUTIL run. When the QUTIL run is registered as a BIS run, it should be allowed access to all modes that may be used for Q-LINK applications development.

3.9 STEP-8: Registering QLINK$DTM

KMSYS Worldwide also provides a BIS run called QLINK$DTM which is required if the BIS DTM interface is to be used in Q-LINK programs. The BIS DTM interface can be configured by setting the "DTM INTERFACE" parameter on the BASIC screen (see Section 3.6.3, "Basic Q-LINK Dynamic Configuration Parameters"). The QLINK$DTM run is in file SYS$LIB$*QLINK. The QLINK$DTM run should be placed in a form type generated with 132-character lines and the full character set. The BIS format option must not be used when retrieving the QLINK$DTM run. When the QLINK$DTM run is registered as a BIS run, it should be allowed access to all modes that may be used for Q-LINK applications development.

3.10 STEP-9: Create TIP VALTAB Entries

All Q-LINK Servers that will be run as TIP transactions must be copied to the appropriate TIP program file and set up in the TIP VALTAB. If you enter the configuration source parameters, "SUPUR FILE CODE FOR TXN SERVERS" and "QLINK VALTAB NAME KEYWORD VALUE", COMUS will automatically perform the absolute copy for you. If are using the ONLINExxxxx files for TIP scheduling, you must perform the copy manually.

There must be two VALTAB entries set up for each unique Q-LINK absolute (BUILD) that will run as a TIP transaction. The same VALTAB can simultaneously serve multiple classes and/or multiple CDBs. VALTAB parameters that are required by Q-LINK are:

NAME

This parameter must be the name of the Q-LINK absolute program. This would generally be the same as the Q-LINK processor name (see the COMUS SRL). If the Q-LINK security feature is utilized and TIP Session Control Security (EXEC) is not enabled, this name (prefixed by a single dollar sign) must be entered as an EXEC user-id on the USER GROUP SGS, which in turn must be referenced on the ACCESS TO UTILITY SGS.

ACTION

This parameter must be the action code specified for the server class name in the class configuration. There may be several action codes for the same program name if you wish to vary parameters such as LEVEL, etc.

Due to the way in which TIP scheduling works, the EXEC will sometimes suspend scheduling of new copies of a "COPIES > 1" transaction. Because of this, Q-LINK requires two VALTAB entries for each program. If scheduling of one ACTION code fails, the other ACTION code will be used. The second ACTION code must be derivable from the first by incrementing the last character of the first ACTION code. For example:

If first ACTION code is:	Then second ACTION must be:
QLINKA	QLINKB
QLT0	QLT1
QLNK9	QLNKA

Because of this quirk of TIP scheduling, it may be desirable to have unique sets of VALTABs for every class using TIP servers, although they will all reference the same Q-LINK absolute (NAME parameter).

AUDIT

If TIP transaction processing is to use MCB and Integrated Recovery, the audit trail number must be set for the application group associated with the server class.

STATUS

If the TIP transaction is to use MCB instead of COMPOOL, the "J" STATUS option must be set.

RUNTIME

Since Q-LINK will monitor the maximum runtime of each request, the runtime specified here should be the highest possible runtime allowed which is 4095. Q-LINK will automatically shutdown and restart a server whose remaining EXEC SUP time is less than its configured Q-LINK maximum SUP time. This will ensure that requests that max-time will be terminated in a controlled fashion.

PRGTYP

Program type is either three (3) for re-entrant, or four (4) for on-line batch. You may use on-line batch to give your Q-LINK Servers a lower memory priority than other TIP transactions. The TIP Priority and Scheduling Level may also be varied.

OPTIONS

"V" (Physical COMPOOL) must ***not*** be set.
"R" (Allow COMPOOL Release) must be set.
"N" (Require Response) must ***not*** be set.

Other options may be used at the discretion of the site.

COPIES

Copies must be greater than or equal to the maximum number of servers allowed in all classes using this VALTAB. Q-LINK will control the number of servers actually started; therefore, a high number should be used here to avoid the necessity of rebuilding the VALTAB to increase the number of allowed servers.

VALTAB examples:

```
766 NAM,QLINK ACT,QLKTA PRG,3 LEV,4 RUN,4095
766 OPT,LRPT STF,0 STA,IJ COP,5 AUD,3
767 NAM,QLINK ACT,QLKTB PRG,3 LEV,4 RUN,4095
767 OPT,LRPT STF,0 STA,IJ COP,5 AUD,3
768 NAM,QLINK ACT,QLKBA PRG,4 LEV,8 RUN,4095
768 OPT,LRPT STF,0 STA,I COP,5
769 NAM,QLINK ACT,QLKBB PRG,4 LEV,8 RUN,4095
769 OPT,LRPT STF,0 STA,I COP,5
```

For additional information regarding TIP VALTAB parameters, refer to the "Transaction Processing and Operations Reference Manual" from Unisys.

Chapter 4: Installation Verification

The Installation Verification Procedure is provided to ensure that all components of Q-LINK have been correctly installed by exercising all interfaces to supporting software.

A Q-LINK Request Procedure named REC-COUNT has been included as a symbolic element in the second product file (SYSLIB*QLINK-1 is the default file name for the default mode install) created by the installation. REC-COUNT is a generalized procedure that may be used to count all records within a DMS 2200 database area by record type. The totals will be sorted by record name and returned to BIS as a result.

Procedure for Executing REC-COUNT:

1. Start Q-LINK as described under the operation chapter of this manual.
2. Sign on to BIS and run the QUTIL BIS run.
3. On the opening menu, enter the name of the Q-LINK server started above and the function letter "C". This will cause QUTIL to display a second screen.
4. On the second screen, enter the following on the blank lines provided:

   INVOKE *subschema-name* IN *schema-name* FILE *qualifier***schema-file-name*

   ADD REC-COUNT FROM SYSLIB*QLINK-1.

   area-name

   <XMIT>

The *subschema-name*, *schema-name*, *schema-file-name*, *qualifier* and *area-name* must be supplied by the local site.

The clause "TIP *TIP-file-number*" may be used instead of the "FILE *file-name*" clause if a TIP schema file is normally used at your site.

Refer to the Q-LINK Programmer Reference for complete descriptions of the INVOKE and ADD directives.

For installation verification, the area to be used should be relatively small, but should include at least three record types to exercise the SORT interface. The time required to complete this process will vary depending upon the size of the area used. An area of 10 pages will be processed in a matter of seconds, while an area of 10,000 pages may take several minutes.

Upon completion of the REC-COUNT request, a sorted list of the names of all record types in the named area and totals for each will be displayed in the current BIS result by the QUTIL run. If this does not occur, check the Q-LINK installation.

Chapter 5: Q-LINK Configurations

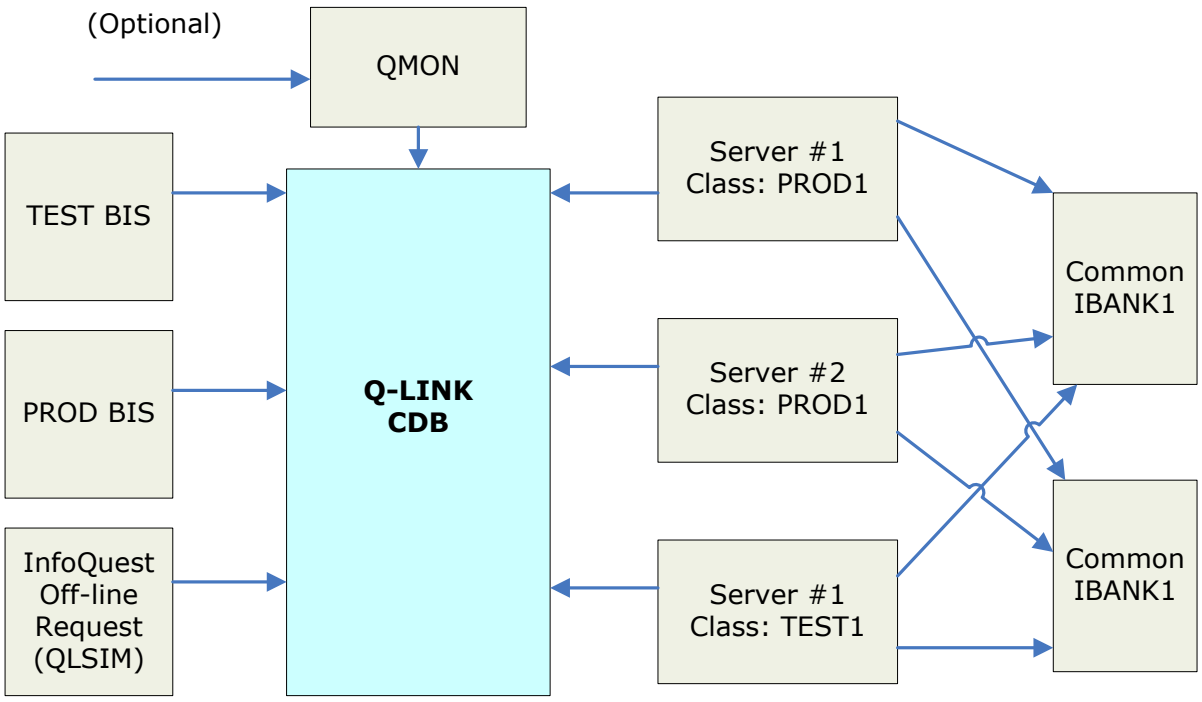
This chapter will illustrate several Q-LINK configurations. One of the configurations may be appropriate for your site; however, KMSYS Worldwide does not recommend that they be installed as shown without first investigating the Q-LINK operational requirements at your site.

5.1 Q-LINK Basic Configuration

The first configuration is meant to illustrate Q-LINK's use of common banks. The common data bank (CDB) is used by all servers executing in the Q-LINK environment regardless of class. The BIS TCF generated during the COMUS INSTALL has been applied to two BIS systems (TEST and PROD) allowing both BISs access to the CDB through the QLK external run function. In addition, the InfoQuest off-line batch processor, QLSIM, will have access to the same CDB. Multiple servers executing in the same class will share common I-Banks.

This configuration represents a single Q-LINK installation (one COMUS REGISTER, BUILD and INSTALL) with one configuration set (one CONFIGURE and PROCESS).

Typical Installation and Configuration



5.2 Multiple Installations

The second configuration illustrates two separate installations of Q-LINK, but utilizing a single common data bank (CDB). The second installation of Q-LINK could provide different features or be of a different size than the first installation. The parameters that dictate which features are to be included, the size of the Q-LINK D-BANK, and the linkage to a particular CDB require a COMUS BUILD and INSTALL. Each installation has its own product file but will share the CDB. This configuration could also be used to introduce a new release level of Q-LINK while maintaining a previous level of Q-LINK.

The first installation of Q-LINK created a product file that will be shared by a second Q-LINK installation. This product file will be referred to as the primary or shared product file. It contains the Q-LINK absolute from the first INSTALL and the configuration element, QLNK$CFIG, that contains the BDI of the CDB that will be shared by both Q-LINK installations. The BDI is used by the QMON processor to initially load the CDB.

The second installation of Q-LINK will create a second product file containing Q-LINK and QLSIM absolutes as a result of using a different installation mode (MODE=A) on the COMUS INSTALL.

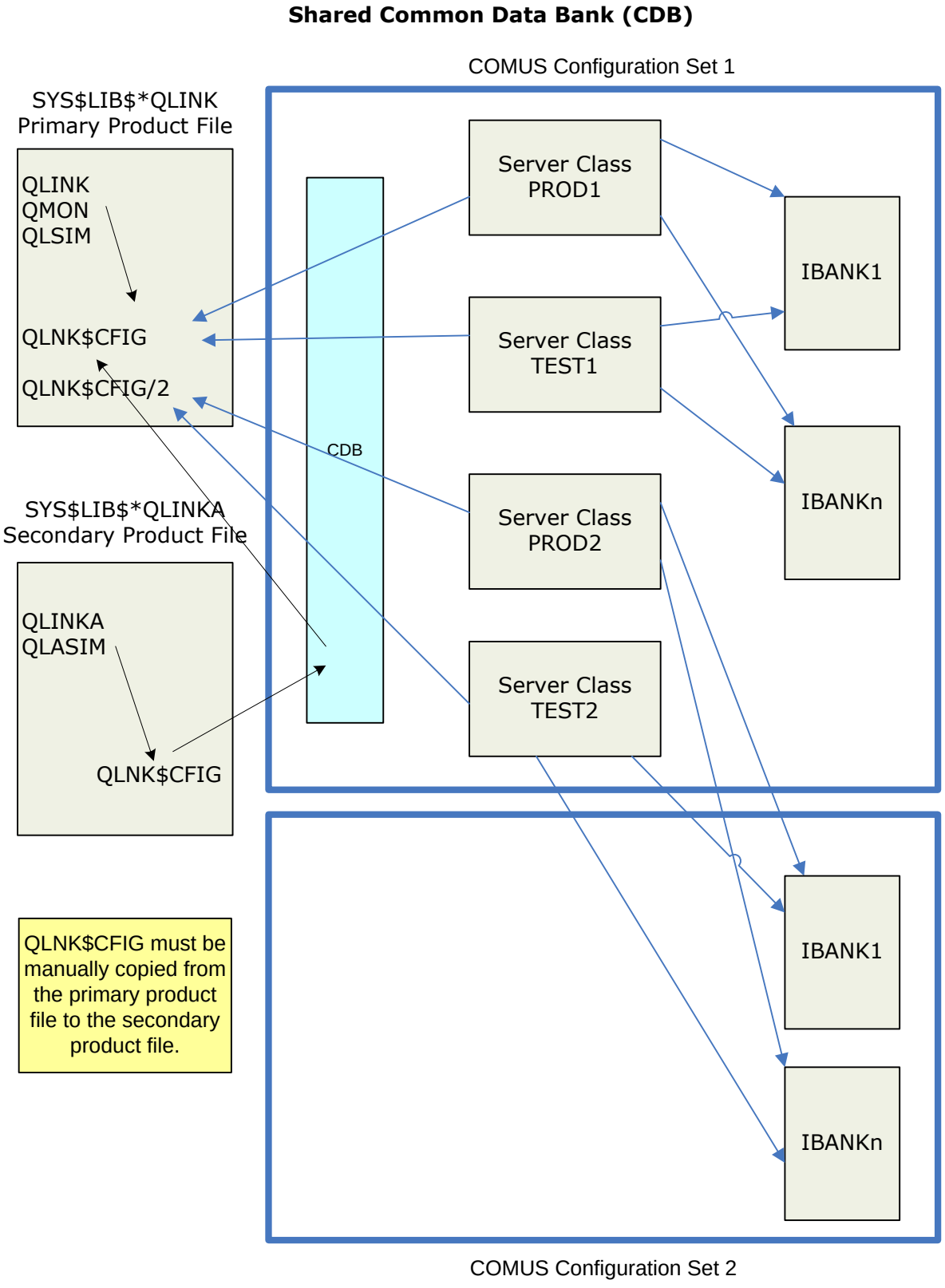
When installing a separate copy of Q-LINK, care must be taken to assign unique BDIs, bank names, processor names and COMUS installation mode. Failing to do so will cause COMUS to de-install the primary installation.

No CDB or QMON processor will be installed into the secondary product file (only one QMON is allowed or needed). The absence of the CDB and the QMON processor results from answering YES to the COMUS INSTALL prompt, "Will QLINKA share an existing QLINK CDB?".

The QLNK$CFIG element from the primary product file will be used to reference the CDB and must be manually copied to the secondary product file once the INSTALL process for the second installation has been completed.

When configuring a second installation, the QLNK$CFIG element needs to be placed into the primary product file for the second configuration. This is accomplished by specifying the primary product file name for the INSTALLATION FILE dynamic configuration parameter. Also, since a QLNK$CFIG element already exists in the primary product file, a unique CONFIGURATION VERSION must also be entered on the same screen. All servers must be configured in the primary product file (Configuration Set 1). The server classes for the second installation must be configured with the same CONFIGURATION VERSION name mentioned above.

This second configuration represents two separate Q-LINK installations (two configuration sets with one BUILD, INSTALL, CONFIGURE, and PROCESS for each).

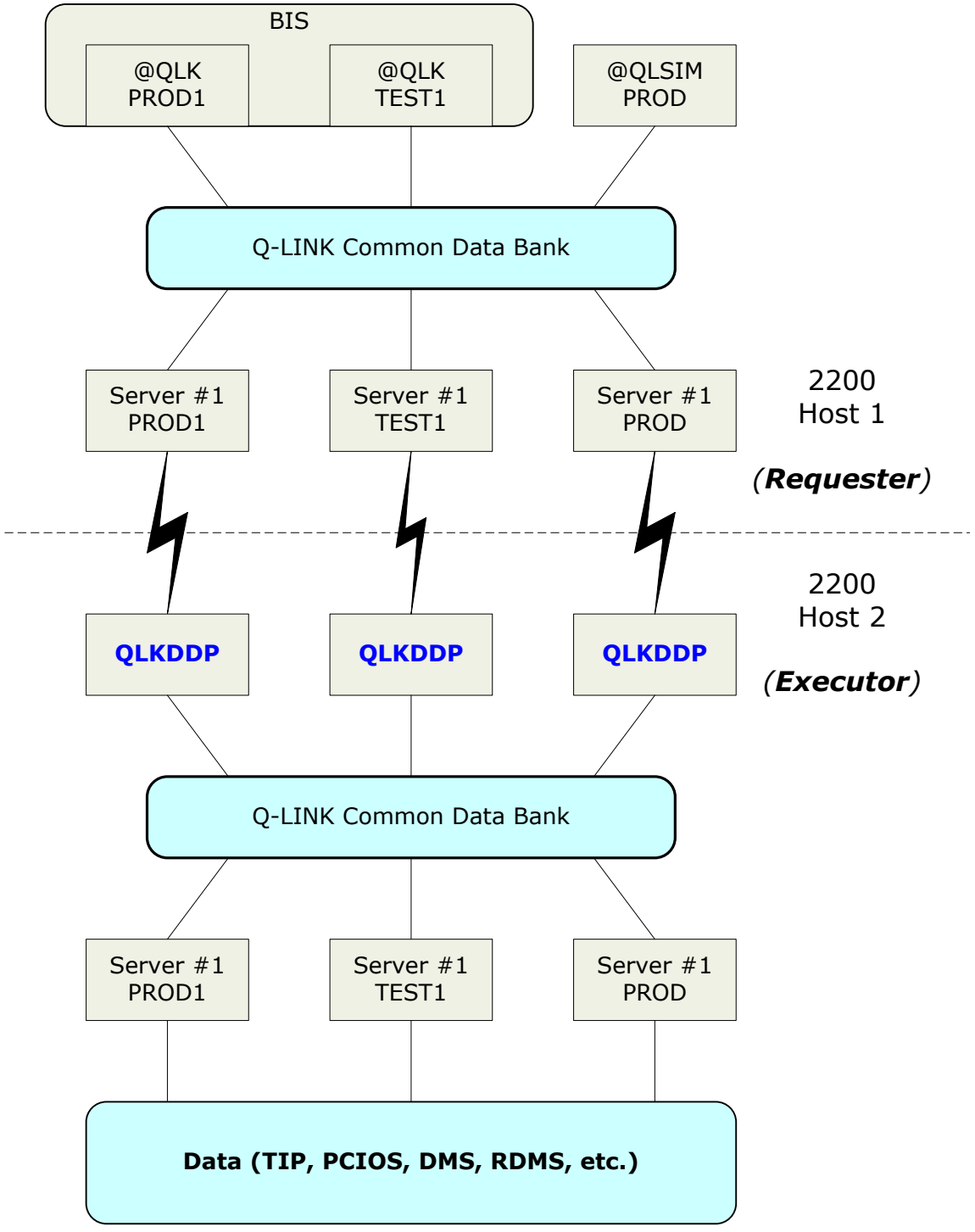


5.3 Q-LINK DDP Configuration

This example shows the Q-LINK DDP-PPC configuration on two 2200 systems. A sample session in this configuration would allow a BIS user on Host 1 to submit a Q-LINK request to server PROD1. By examining the @QLK parameters, the server PROD1 recognizes that the submitted request is to be passed to Host 2 for actual execution. The server PROD1 connects to the DDP-PPC facility that causes the QLDDP run to execute on Host 2.

In this example, the Q-LINK software was installed using a "MODE=K" install in COMUS. During the "MODE=K" install, the QLDDP absolute name was changed to QLKDDP. The QLKDDP program starts and connects to DDP-PPC on Host 2. The Q-LINK request from Host 1 is sent (via QLKDDP) to the equivalent PROD1 server on Host 2 for execution. The result of the Q-LINK request is returned to the PROD1 server on Host 1 via QLKDDP. After a period of inactivity, the QLKDDP program will terminate automatically. Note that compiled Q-LINK programs executed via the RUN directive will be loaded and executed on the executor system and not from the requester system. Please see the Q-LINK Programmer Reference Manual for additional programming considerations when using the remote execution facility.

Multiple Hosts using DDP-PPC



Chapter 6: Security

SGSs are provided to control security at your site in a variety of ways. Access can be ALLOWED or DENIED to specific users or by designated user groups. The following subsections allow for security control to Q-LINK utility programs, DMS 2200 schemas, subschemas, and DMRs; PCIOS files; DIO files; RDMS 2200 files and BIS DTM queues.

6.1 Impact on Execution

Changes made to the security SGSs do not require a Q-LINK generation or installation (COMUS BUILD/INSTALL). Once changes have been made to the element containing the security SGSs, the COMUS CONFIGURE PROCESS must be used for the changes to take effect.

After the COMUS CONFIGURE PROCESS has been completed successfully, the Q-LINK common data bank (CDB) must be reloaded with the QMON INIT command if changes were made to the UTILITY program security configuration.

6.2 Pre-scanning Security SGSs

For sites wishing to pre-scan their Q-LINK Security SGSs for proper syntax prior to doing a COMUS CONFIGURE PROCESS, the following SSG runstream may be used:

```
@SSG,A SYS$LIB$*QLINKn.SECSKEL,q*f.yoursecurity/SSGs
@EOF
```

Where "QLINKn" is the Q-LINK Installation File as configured in COMUS.

This procedure will make it possible to verify the SGSs without having to take Q-LINK out of the production environment. It can also result in a substantial savings of time, as the COMUS CONFIGURE PROCESS must reprocess a number of elements in addition to the one for security.

6.3 General Notes on Security Groups

All of the group names below must be 1 to 12 characters in length. Valid names may be comprised of uppercase letters ("A" through "Z"), lowercase letters ("a" through "z"), the underscore character ("_") and the dollar sign ("$"). In addition, names may not begin with a dollar sign or underscore.

The names used in describing the member(s) of a group can be either the actual name of the object (i.e., a schema name) or can be truncated with a trailing "!" character. If the name is truncated in this manner, it will match any candidate name beginning with the

same sub-string before the "!". In other words, MIS! would match MIS-SCHEMA1, or MIS-SCHEMA2, or MIS.

Wherever a group name is required on a security image, the pseudo name, $ALL, may be used to indicate that any group is allowed or denied access. For example, let us assume that only one user group ("QLMAINT") is to be allowed access to certain server classes ("BIGT" and "DEV1") configured with Q-LINK. However, all users can have access to all other servers configured. The security SGSs might appear as follows:

```
USER GROUP QLMAINT HAS DAVID JEAN
CLASS GROUP RESTRICTED HAS BIGT DEV1
...
ACCESS TO CLASS RESTRICTED ALLOWED FOR QLMAINT ;
FOR $ALL
ACCESS TO CLASS RESTRICTED DENIED FOR $ALL
ACCESS TO CLASS $ALL ALLOWED FOR $ALL $ALL
```

In the example, users JEAN or DAVID would be allowed access to the server classes BIGT and DEV1 (grouped together in the class group called RESTRICTED). Another user whose user-id is THOMAS (not shown) would be DENIED access to either of these classes since THOMAS falls into the global user group $ALL and is not a member of user group QLMAINT. However, THOMAS would be ALLOWED access to all other classes since the classes fall into the global class name $ALL.

$ALL may be used in clauses other than group names where shown in the syntax.

IF security is enabled for any particular facility (e.g., access to a DMR), there must be an ACCESS ALLOWED SGS for all other facilities required (e.g., utilities such as QINDEX or file interfaces such as PCIOS).

6.3.1 USER GROUP

This security image defines groups of users who can be allowed or denied access to a given facility by Q-LINK.

Format:

USER GROUP *user-group* HAS || *user-id-1* | *user-id-2* | … *user-id-n* ||

For most uses of this security image, *user-id-1* through *user-id-n* are the BIS user-ids; however, the ACCESS TO UTILITY image requires that separate user groups be defined since the utilities are run outside BIS and require EXEC user-ids. In addition, when TIP servers are used, the Q-LINK absolute program name must be prefixed by a dollar sign ($) and entered as an EXEC user-id.

It is possible to lock yourself out of Q-LINK by having only invalid or unusable user-ids in the security SGS list. If this happens you will not have access to QMON and the Q-LINK common data bank (see Section 7.4.3, "Reinitializing the CDB Security").

6.4 Security on Q-LINK Utilities

The following parameters control access to the Q-LINK utilities by user groups.

6.4.1 UTILITY TYPE

This security image defines groups of Q-LINK utility programs, which will be named later on the ACCESS TO UTILITY SGS.

Format:

```
UTILITY TYPE utility-group HAS {|| QLINK | ;
                                   QINDEX | ;
                                   QMON | ;
                                   QLSIM | ;
                                   MAPPER || ;
                                | $ALL}
```

6.4.2 ACCESS TO UTILITY: ALLOWED

The ACCESS TO UTILITY SGSs allow control of which users can use any or all of the utility programs that comprise the Q-LINK product. This parameter is useful in order to ensure control of who might execute a sensitive program such as QMON. An optional clause is provided to limit access to privileged commands. If the clause is omitted, or specifies LIMITED access, privileged functions will be disabled for the specified user-id(s). Currently this clause controls access to QMON for commands other than status type functions.

Format:

```
ACCESS TO UTILITY utility-group ALLOWED ;
   FOR EXEC-user-group [IS {LIMITED | UNLIMITED} ]
```

ACCESS TO UTILITY controls who can run the Q-LINK utilities (see "UTILITY TYPE" above), ***not*** which BIS user can access Q-LINK (i.e., the servers). Access to the servers is controlled by the ACCESS TO CLASS...SGS.

6.4.3 ACCESS TO UTILITY: DENIED

This parameter is the inverse of the ALLOWED parameter and is useful if your configuration has fewer denied combinations than allowed combinations. Since the DENIED image always causes the access check to fail, there is no need to specify the allowed usage modes here.

Format:

```
ACCESS TO UTILITY utility-group DENIED FOR EXEC-user-group
```

6.5 Q-LINK Security on Server Classes

Access to specific server classes can be limited by class name. In addition, a class group can be limited to either compiling or executing Q-LINK programs, if desired.

6.5.1 CLASS GROUP

This security image defines groups of class names that will be named later on the ACCESS TO CLASS parameter cards. A group can have as many members (class names) as are configured in Q-LINK. There can be as many or as few groups as are useful to your security requirements.

Format:

CLASS GROUP *class-group* HAS || *class-name-1* | ;
class-name-2 | ;
… *class-name-n* ||

6.5.2 ACCESS TO CLASS … ALLOWED

This security image controls access to the server class names dynamically configured within Q-LINK.

Format:

ACCESS TO CLASS class-group ALLOWED ;
FOR *BIS-user-group* FOR {COMPILE | XQT | $ALL}

6.5.3 ACCESS TO CLASS … DENIED

This parameter is the inverse of the ALLOWED parameter and is useful if your configuration has fewer denied combinations than allowed combinations. Since the DENIED image always causes the access check to fail, there is no need to specify the allowed usage modes here.

Format:

ACCESS TO CLASS *class-group* DENIED ;
FOR *BIS-user-group*

6.6 Q-LINK Security for DMS 2200 Access

The following parameters control access to the DMS 2200 facility by schemas, subschemas, schema files and DMRs. The interpretation of these images is fully explained in the "Search Algorithm for Security (DMS)," below.

6.6.1 SCHEMA GROUP

This security image defines groups of schema names, which are named subsequently on the ACCESS security images. A group can have as many members (schema names) as desired. A trivial group consists of only one schema name. There can be as many or as few groups as are useful to your security requirements.

Format:

```
SCHEMA GROUP schema-group HAS || schema-name-1 | ;
                                 schema-name-2 | ;
                               … schema-name-n ||
```

6.6.2 SUBSCHEMA GROUP

This security image defines groups of subschemas, which are named subsequently on the ACCESS security images. A group can have as many members as desired. A trivial group consists of only one subschema name. There can be as many or as few groups as are useful to your security requirements.

Format:

```
SUBSCHEMA GROUP subschema-group HAS || subschema-name-1 | ;
                                      subschema-name-2 | ;
                                    … subschema-name-n ||
```

6.6.3 SCHEMAFILE GROUP

This security image defines groups of schema files that can be associated with a given DMR. You may need multiple groups for a given DMR if only some users are to be allowed impart access to a given schema file.

Format:

```
SCHEMAFILE GROUP schemafile-group ;
   HAS || {FILE,qual*filename-1 | TIP,file-code-1} | ;
          {FILE,qual*filename-2 | TIP,file-code-2} | ;
        … {FILE,qual*filename-n | TIP,file-code-n} ||
```

6.6.4 ACCESS TO DMR … ALLOWED

This security image controls access to the DMS 2200 facility within Q-LINK. The optional INVOKING fields allow control of what subschemas a user can access at IMPART time. The default schema and subschema for a given DMR is set as part of the configuration of that DMR.

Format:

```
ACCESS TO DMR {DMR-name | $ALL} ALLOWED FOR BIS-user-group ;
   FOR {RETRIEVAL | UPDATE | PW_UPDATE | LOAD} ;
   [INVOKING subschema-group OF schema-group FILE schemafile-group]
```

The access types (RETRIEVAL, UPDATE, PW_UPDATE and LOAD) are listed in the order of privilege; i.e., UPDATE is higher than RETRIEVAL, PW_UPDATE is higher than RETRIEVAL or UPDATE, and LOAD is higher than the other three. For example, if you have UPDATE

privileges you can open an area for RETRIEVAL and do not need to specify an SGS allowing RETRIEVAL privileges.

Opening areas for LOAD from Q-LINK is not recommended.

6.6.5 ACCESS TO DMR … DENIED

This parameter is the inverse of the ALLOWED parameter and is useful if your configuration has fewer denied combinations than allowed combinations. Since the DENIED image always causes the access check to fail, there is no need to specify the allowed usage modes here.

Format:

ACCESS TO DMR {*DMR-name* | $ALL} DENIED FOR *BIS-user-group*

6.6.6 DEFAULT DMR

This image sets the default DMR for the INVOKE directive when multiple DMRs are configured in a single Q-LINK. This is useful when merging groups of users into a single Q-LINK. This provides source level compatibility for their INVOKE statements to select the same DMR they would have accessed under separate Q-LINK configurations.

Format:

DEFAULT DMR IS *DMR-name* For { || *BIS-user-group-1* | ;
BIS-user-group-2 | ;
… *BIS-user-group-n* || ;
| $ALL}

Q-LINK normally selects the first multi-thread as the default DMR. This image will override that default for any users named in the user group list. The pseudonym, $ALL, can be used to set a global default DMR. These images are searched serially with the first match condition being used; therefore, selected user groups can be assigned to selected DMRs and then a global $ALL used to assign all others to a specific DMR (see Section 6.6.7, "Search Algorithm for Security (DMS)"). Unlike the security images, a "no find" condition does not cause rejection of the request, only a reversion to Q-LINK's normal default DMR selection.

6.6.7 Search Algorithm for Security (DMS)

The security information is searched sequentially looking for an indication of access being allowed to a facility. If no explicit access is allowed, then access will not be permitted by Q-LINK. Two basic strategies can be used in building the security tables:

1. You could list all the ALLOWED combinations and then permit the undeclared to fail the search, and be rejected.
2. Alternately, you could list the DENIED combinations and then terminate the list with:

   ```
   ACCESS TO DMR $ALL ALLOWED FOR $ALL
   ```

   This image would allow the user to INVOKE/IMPART to anything for anything if they were not specifically denied access previously. Many other variations are possible depending on your site's DMR configurations and security requirements.

There are two points at which DMS security information is checked. The security tables are searched at IMPART time. The ACCESS image that allowed the user to IMPART is saved and the access modes from that image are checked whenever the user wishes to OPEN an area.

The security tables are searched only on a feature (e.g., DMS) and object (e.g., user, DMR name, schema file, schema and subschema) relationship. The access type is NOT a search criteria, it is the result returned if the table search indicates that some access is allowed. In other words, if a find occurs indicating that only retrieval type access is allowed, it does not

matter that a later ACCESS image would allow this object combination to update, an OPEN FOR UPDATE would be rejected.

Note that there is no security enforcement of what a user can compile. He can INVOKE, DEFINE F, or compile any syntactically correct image (subject to EXEC access control on the named schema file). Security enforcement is performed ONLY at run time and only if the command is executed. A program could contain legal and illegal commands for a given user and if the flow of control never encountered an illegal command, the program could execute normally.

6.7 Q-LINK Security for PCIOS Access

These parameter cards control access modes for PCIOS files by file type. The interpretation of these images is similar to that of the DMR facility.

6.7.1 PCIOS TYPE

This parameter controls the type of PCIOS file to be accessed by a PCIOS group.

Format:

PCIOS TYPE *PCIOS-group* HAS { || CATALOGED | TEMPORARY || | $ALL}

Both CATALOGED and TEMPORARY can be specified for the same PCIOS group.

6.7.2 ACCESS TO PCIOS … ALLOWED

This security image controls access to the PCIOS facility within Q-LINK.

Format:

ACCESS TO PCIOS *PCIO-group* ALLOWED FOR *BIS-user-group* ;
FOR {READ | READ_WRITE | $ALL}

6.7.3 ACCESS TO PCIOS … DENIED

This is the inverse of the ALLOWED parameter and is useful if your configuration has fewer denied combinations than allowed combinations. Since the DENIED image always causes the access check to fail, there is no need to specify the allowed usage modes here.

Format:

ACCESS TO PCIOS *PCIOS-group* DENIED FOR *BIS-user-group*

6.7.4 Search Algorithm for Security (PCIOS)

The search of the security records will be performed at file OPEN time. The file's OPEN request (INPUT, OUTPUT, UPDATE, etc.) will be compared to the permissions granted to that user. If it is valid, no further checks will be performed. A problem could exist if the user changes the file name association via CSF$ @USE after the OPEN (not logical, but someone could do it).

6.8 Q-LINK Security for DIO Access

These parameter cards control access modes for DIO files by file type. The interpretation of these images is similar to that of the DMR facility.

6.8.1 DIO TYPE

This SGS controls the type of DIO file to be accessed by a DIO group.

Format:

```
DIO TYPE DIO-group HAS { || FCSS | ;
                          TIPDMS | ;
                          CATALOGED | ;
                          TEMPORARY || ;
                       | $ALL}
```

A combination of FCSS, TIPDMS, CATALOGED, and TEMPORARY can be specified for a DIO group.

6.8.2 ACCESS FOR DIO ... ALLOWED

This security image controls access to the DIO facility within Q-LINK.

Format:

```
ACCESS TO DIO DIO-group ALLOWED FOR BIS-user-group ;
   FOR {READ | READ_WRITE | $ALL}
```

6.8.3 ACCESS TO DIO ... DENIED

This parameter is the inverse of the ALLOWED parameter and is useful if your configuration has fewer denied combinations than allowed combinations. Since the DENIED image always causes the access check to fail, there is no need to specify the allowed usage modes here.

Format:

```
ACCESS TO DIO DIO-group DENIED FOR BIS-user-group
```

6.8.4 Search Algorithm for Security (DIO)

The search of the security records will be performed on the first I/O via the specified FCT. The program can (and frequently will) change the file association via @USE and we cannot economically detect this change. This problem only exists if you allow direct I/O to temporary or cataloged files but not to both, for a given user. Otherwise, the system is solid. Realistically, there is not much security if you split the enforcement of temporary and cataloged files other than that provided by the EXEC and with site programming standards.

6.9 Q-LINK Security for RDMS Access

Since UDS provides most security controls required for RDMS, Q-LINK does not duplicate that effort. The only security configurable in Q-LINK is a simple check as to whether a user is allowed to access a given RDMS application. Detail control of access and update privileges is left to UDS. The only security provided for RDMS 2200 access is whether a user can access RDMS. Any additional security desired should be configured through the data dictionary interface for UDS.

6.9.1 ACCESS TO RDMS ... ALLOWED

This security image controls access to the RDMS 2200 facility within Q-LINK.

Format:

ACCESS TO RDMR {*application*-name | $ALL} ALLOWED FOR *user-group*

6.9.2 ACCESS TO RDMS ... DENIED

This parameter is the inverse of the ALLOWED parameter and is useful if your configuration has fewer denied combinations than allowed combinations.

Format:

ACCESS TO RDMR {*application*-name | $ALL} DENIED FOR *user-group*

6.10 Q-LINK Security for DTM Access

Since BIS provides most security for DTM access, Q-LINK does not duplicate that effort. The Q-LINK security feature can be configured to control which users are allowed to access which BIS queue names and whether they can read and write, or just read reports. Note that the user must still supply a valid BIS sign-on in the parameter block and that the sign-on must be validated for access to the run specified in the parameter block and that run must be allowed access to the requested report.

The ACCESS TO DTM parameters allow control of which users can access a particular BIS DTM SCHDLR queue name and whether it may be opened for INPUT or OUTPUT (the only currently supported access modes). This security is in addition to the security imposed on the DTM interface by BIS in requiring a valid user-id/department/password combination along with the run registration security of what reports can be accessed.

6.10.1 ACCESS TO DTM … ALLOWED

This security image controls access to the DTM facility within Q-LINK.

Format:

ACCESS TO DTM {*queue-name* | $ALL} ALLOWED FOR *BIS-user-group* ;
FOR {READ | READ_WRITE | #ALL}

6.10.2 ACCESS TO DTM … DENIED

This parameter is the inverse of the ALLOWED parameter and is useful if your configuration has fewer denied combinations than allowed combinations. Since the DENIED image always causes the access check to fail, there is no need to specify the allowed usage modes here.

Format:

ACCESS TO DTM {*queue-name* | $ALL} DENIED FOR *user-group*

6.11 Security Examples

The following example illustrates a security configuration DMS, PCIOS and DIO access. If no security SGSs were supplied, then there is no security. If security is desired for one file structure, then security SGSs must be supplied for all file structures (see the PCIOS SGSs in the example below).

The server class SGSs allow only the TESTALL BIS user group to use the TEST1 server class. User group XQTONLY can access the PROD1 server class through the PRODSERVER class group, but only for executions; however, they (DAVIS, GEOFF, ANDY and LARRY) may compile as well as execute by using the OFFLIN server; i. e., they would fit into the $ALL user group for the INFOQUEST class group. Any other BIS user may compile and execute using either the PROD1 or OFFLIN server classes.

The DMS security SGSs are for a system with two DMRs; APPL1 is the production DMR and APPL2 is the test DMR. There are two production schemas running under APPL1: HRDB and MFGFINDB. A test version of these schemas runs under APPL2. There are two subschema groups that are allowed to access the HRDB schema only: PAYROLL and PLANNING. The ACCTG and ORDENTRY subschema groups are allowed only to access the MFGFINDB schema.

All production schemas and subschemas reside in TIP file 107, while the test versions are in EXEC file DMS2*SCHFILE.

The user groups QLMAPSTAFF and SECURITY will INVOKE against the APPL2 DMR if they do not specify the DMR on the INVOKE directive. However, they are not restricted only to using the APPL2 DMR.

Notice that no user can access either DMR for LOAD. Opening areas for INITIAL LOAD from Q-LINK is not recommended.

Only the users listed in user groups OPERATIONS and QLEXECSTAFF have "UNLIMITED" access to the QMON utility. All other user access to the QMON utility is "LIMITED" to status type keyins only.

As described earlier, the ACCESS SGS can be used to restrict access. One example of this is the ACCESS SGS for the HRQUERY user group shown below. They are restricted to accessing only the PAYROLL subschema group, which has two subschemas grouped together: PAYSUB and PERSUB. Furthermore, they can only use the HRDB schema in TIP file 107.

The security shown below for DIO is an example of how to restrict access utilizing one of the "ACCESS...DENIED" SGSs. Only the users in the SECURITY users group can use DIO commands on all file types. Any other user can use DIO commands on CATALOGED and TEMPORARY files only.

Example:

```
.  Security SGSs for Q-LINK Utilities — Requires EXEC User-id.
USER GROUP OPERATIONS HAS BATCHSYS, SECURI
USER GROUP TIP HAS $QLINK
USER GROUP QLEXECSTAFF RENEEDP HANSDP
.
UTILITY TYPE TXN HAS QLINK
UTILITY TYPE BAT HAS QMON QLINK QLSIM BIS
UTILITY TYPE QLMONITOR HAS QMON
UTILITY TYPE QLINDEX HAS QINDEX
ACCESS TO UTILITY QLMONITOR ALLOWED FOR QLEXECSTAFF ;
    IS UNLIMITED
```

```
ACCESS TO UTILITY QLMONITOR ALLOWED FOR $ALL ;
    IS LIMITED
ACCESS TO UTILITY TXN ALLOWED FOR TIP
ACCESS TO UTILITY BAT ALLOWED FOR OPERATIONS
    IS UNLIMITED
ACCESS TO UTILITY QLINDEX ALLOWED FOR $ALL
.
.-Security SGSs for Q-LINK Access — Requires BIS User-ids.
USER GROUP HRENTRY HAS CLAUSE ROBERT LINDA
USER GROUP HRQUERY HAS DAVID GEOFF
USER GROUP RESMGT HAS JEAN GEOFF2
USER GROUP ACCTUSERS HAS ANDIE CARLOS JANE
USER GROUP OEUSERS HAS ANDY LARRY
USER GROUP QLMAPSTAFF HAS RENEE HANS
USER GROUP SECURITY HAS COORD
USER GROUP TESTALL HAS RENEE HANS COORD
USER GROUP XQTONLY HAS DAVID GEOFF ANDY LARRY
.
.
CLASS GROUP TESTSERVER HAS TEST1
CLASS GROUP PRODSERVER HAS PROD1
CLASS GROUP INFOQUEST HAS PROD1 OFFLIN
ACCESS TO CLASS TESTSERVER ALLOWED ;
    FOR TESTALL FOR $ALL
ACCESS TO CLASS PRODSERVER ALLOWED ;
    FOR XQTONLY FOR XQT
ACCESS TO CLASS INFOQUEST ALLOWED ;
    FOR $ALL FOR $ALL
.
.
DEFAULT DMR IS APPL2 ;
    FOR QLMAPSTAFF SECURITY
SCHEMA GROUP PRODUCTION HAS HRDB MFGFINDB
SCHEMA GROUP TEST HAS HRDB MFGFINDB NEWDB
SUBSCHEMA GROUP PAYROLL HAS PAYSUB PERSUB
SUBSCHEMA GROUP PLANNING HAS PERSUB PROJSUB
SUBSCHEMA GROUP ACCTG HAS APSUB ARSUB
SUBSCHEMA GROUP ORDENTRY HAS INVSUB ARSUB
SCHEMAFILE GROUP ONLINE HAS TIP,107
SCHEMAFILE GROUP OFFLINE HAS FILE,DMS2*SCHFILE
.
ACCESS TO DMR APPL1 ALLOWED FOR HRENTRY ;
    FOR UPDATE ;
    INVOKING PAYROLL OF PRODUCTION FILE ONLINE
ACCESS TO DMR APPL1 ALLOWED FOR HRQUERY ;
    FOR RETRIEVAL ;
    INVOKING PAYROLL OF PRODUCTION FILE ONLINE
ACCESS TO DMR APPL1 ALLOWED FOR RESMGT ;
    FOR UPDATE ;
    INVOKING PLANNING OF PRODUCTION FILE ONLINE
ACCESS TO DMR APPL1 ALLOWED FOR ACCTUSERS ;
    FOR UPDATE ;
```

```
    INVOKING ACCTG OF PRODUCTION FILE ONLINE
ACCESS TO DMR APPL1 ALLOWED FOR OEUSERS ;
    FOR UPDATE ;
    INVOKING ORDENTRY OF PRODUCTION FILE ONLINE
ACCESS TO DMR APPL1 ALLOWED FOR QLMAPSTAFF ;
    FOR UPDATE
.
ACCESS TO DMR APPL2 ALLOWED FOR QLMAPSTAFF ;
    FOR UPDATE
ACCESS TO DMR $ALL ALLOWED FOR SECURITY ;
    FOR UPDATE
.
.
ACCESS TO PCIOS $ALL ;
ALLOWED FOR $ALL FOR $ALL
.
.
DIO TYPE SECURE HAS FCSS TIPDMS
ACCESS TO DIO $ALL ALLOWED FOR SECURITY ; FOR READ_WRITE
ACCESS TO DIO SECURE DENIED FOR $ALL
ACCESS TO DIO $ALL ALLOWED FOR $ALL FOR $ALL
```

Chapter 7: Q-LINK Operation

This chapter contains the information necessary to initialize, control and monitor Q-LINK. It covers the handling of the Q-LINK Common Data Bank (CDB), the management of servers and the monitoring of performance utilizing the QMON processor.

7.1 Technical Overview

Q-LINK allows BIS runs to invoke DMS 2200 subschemas, open PCIOS files, access TIP files, open RDMS 2200 tables, access BIS data via DTM and process in any or all of these environments, using all of the security and recoverability features inherent in the native mode of each environment. This activity does not actually take place in BIS. It is performed by separate programs linked to BIS through a common data bank.

The Q-LINK operating environment consists of three major components: a BIS run function called QLK; the Q-LINK server programs; and QMON, the Q-LINK operations interface program. QLK is a BIS run function supplied with BIS (for BIS levels earlier than 35, this function is supplied with Q-LINK) which communicates with Q-LINK servers. When called from a BIS run, QLK will read the RID containing the Q-LINK request and pass input (the request) to a designated Q-LINK server. The request consists of procedural commands and directives, which are coded using the Q-LINK programming language, and input data required by the procedure. The request will direct the server module in such activities as compiling, saving, loading and executing Q-LINK procedures. Data returned to BIS by the procedure is written to a result RID by QLK.

The line of communication established between QLK and the server module is a two-way path. Once the communication session between Q-LINK components is established, input data (the request) and output data (the reply) can pass between the BIS and non-BIS environments as required by the user written Q-LINK procedure. This two-way communication path facilitates efficient data exchange in both the reading of large BIS reports, and in the creation of result reports.

7.2 Common Data Bank

As stated earlier, BIS is linked to the Q-LINK server through a common bank. The Q-LINK common bank contains the current runtime configuration, and all the tables and code necessary to control the interaction of BIS and Q-LINK. This includes the automatic routing of requests received from BIS to various Q-LINK server programs.

Each active server program belongs to a server class, and there may be multiple servers active in a class. Server class registration is controlled by the Q-LINK runtime configuration. Each active server has a server control table assigned to it within the common bank. The transfer of data between an individual server module and a BIS activity

is completely independent of other Q-LINK requests, which may be processing simultaneously.

7.3 The Q-LINK Server

The Q-LINK server is the system component that processes the user coded Q-LINK program. The actual communication between BIS and a Q-LINK server program is accomplished by the QLK function and the server activating one another via the Q-LINK common bank.

When a Q-LINK server is started, control information is posted to the common bank. The server will then be placed in an idle state. The server is activated when a BIS request is routed to it for processing. QLK will pack data into a transfer buffer and notify the CDB of the address and size of the buffer. The data is moved to the server's data storage. The server is then activated, and parsing and execution of the Q-LINK program begins.

As program execution proceeds, the QLK function has been suspended waiting for the server to activate it. Output from the server to BIS is packed in a data transfer buffer. When the transfer buffer is filled, it is passed to the CDB where the data is transferred to the BIS buffer and the QLK function is activated. QLK will open the result RID (if not already open) and write the report lines.

This process continues until the Q-LINK request is complete. When the QLK function is notified that the request is complete, the result report is closed and the BIS run continues. The Q-LINK server is once again placed in an idle state, waiting for another request.

Multiple Q-LINK servers may be active simultaneously. Each server belongs to a server class, and multiple server programs may be active in the class. The QLK function called in the BIS run will specify a server class name to be used in processing a particular request. Q-LINK will locate the first idle server belonging to the specified class and will assign the BIS request to this server. For the duration of the request, the assigned server module is "owned" by the requesting BIS run. At the completion of the request, the server is marked as available and placed in an idle state. If all servers in a class are busy when a request is received, the incoming request is queued for the specified class and an additional server is started if this is configured. The first server in the requested class to become idle will immediately have the queued request routed to it for processing. An idle server module does not execute any code. The activity is queued by the EXEC while waiting to be activated by a requesting BIS run.

The server program can be either a batch or a TIP transaction. Server priority is established by the @RUN image for batch servers or by the VALTAB for TIP transaction servers. Servers within the same class should operate at the same priorities since server selection within a class is not predictable.

7.3.1 Server File Usage

Before Q-LINK is brought up for the first time, the default object procedure file and the default source procedure file must be cataloged by the Q-LINK administrator. When a Q-LINK procedure issues the RUN directive without having sent the procedure as part of the request, Q-LINK will load the named procedure from an object procedure file. The SAVE directive will notify Q-LINK that the procedure being sent is to be saved in a procedure source file. Refer to the Q-LINK Programmer Reference for a complete discussion of these directives.

The object and source procedures may be cataloged as follows:

```
@CAT,P Q$LNK*Q$LNKOBJ.  .  - The default object file.
@CAT,P Q$LNK*Q$LNKLIB.  .  - The default source file.
```

If you wish to use separate object libraries for some classes, you may change the qualifier for these classes. See the "Q-LINK LIBRARY QUALIFIER" parameter on the appropriate Server Class configuration screen.

These files are program element files and should be sized by each site based on the number and size of Q-LINK programs that may be saved in source and object form. The object procedure file will be assigned by all active servers. If cache disk is available, it may be desirable to place the object procedure file on cache. It is important to monitor the object procedure file, expanding and packing it as necessary. Any non-default source and object files will be automatically assigned and freed as necessary.

In addition to the source and object files, Q-LINK servers will automatically assign temporary sort work files for using the system sort subroutine interface. Sort work files will be assigned as follows:

```
@ASG,T XA.,F///1000
@ASG,T XB.,F///1000
@ASG,T XC.,F///1000
```

These file assignments can be overridden by the user-written Q-LINK request program by using the CSF command.

Any user file assignments must be controlled by the user request program. Any files not freed by the user request program, will automatically be freed by the Q-LINK processor at the end of request processing (except for the XA, XB, XC and other miscellaneous Q-LINK internal files.

If files are to remain assigned to the server, they should be given an internal file name of QLINKUSE0 through QLINKUSE9 in the Q-LINK program. For example:

```
CSF X '@USE QLINK$USE$0.,QUALIFER*FILENAME.'
```

7.3.2 Initializing Q-LINK

Before any Q-LINK requests can be run the common data bank (CDB) must be loaded and configured. Any successful QMON execution will accomplish this. In particular, "@QMONx,L" will accomplish this without operator intervention. Once this has been accomplished, Q-LINK has been initialized. A number of situations can necessitate the loading of the CDB; e.g., a change to any of the Q-LINK dynamic configuration parameters through the COMUS CONFIGURE PROCESS, any system reboot, a reload request for the CDB, or a fatal error being encountered by the Q-LINK CDB control software.

7.3.2.1 BATCH Servers

The batch server runstreams contain the necessary QMON execution to automatically load the common data bank (CDB) and subsequently start the Q-LINK servers (see Section 7.3.3, "Starting Q-LINK Servers").

7.3.2.2 TIP Servers

When using TIP servers, there are no runstreams to load the common data bank (CDB) automatically; hence, there is no QMON execution performed. For this reason, your operational procedures must assure separate initialization of the CDB. Some suggestions:

- Place the QMON execution in the BIS start-up runstream which will cause Q-LINK to be initialized whenever you bring up BIS (Note: You must @FREE SYSLIB*QLINKx after the QMON processor call or you cannot PROCESS configurations until BIS is down), or
- Start the QMON execution from the EXEC's SYS run, or
- Have the operator can start the batch QMON run after a system boot and before users are allowed access to the system, etc.

When running TIP servers through the MCB, QMON must be executing for TIP servers to start properly; therefore, requiring a separate procedure to initialize the CDB is not required.

7.3.2.3 TIP Servers Accessing MCB vs. COMPOOL

The effective use of Q-LINK servers in an MCB environment requires some planning. Since any given program (batch or transaction) can associate itself with only one application group during any execution, it is important to prevent Q-LINK servers from trying to cross application group boundaries.

A single Q-LINK absolute is capable of accessing TIP through COMPOOL or any configured MCB. This capability will allow the site to maintain only one absolute for use with all application groups. It is, however, necessary to have any given server class associated, by convention, with a single application group. If MCB scheduling instead of COMPOOL scheduling is used, then the application group number will appear in the AUDIT parameter on the VALTAB entry for its server class.

Application group names and MCB entry points must be configured on the GLOBAL and the GLOBALEXT screens to permit a Q-LINK server to access the MCB.

There must be one copy of QMON running for each application group that is to be accessed by MCB scheduled Q-LINK servers.

MCB scheduling is required whenever a site wishes to use transaction servers and COMPOOL is not configured in the EXEC (one such instance is when TIP security is configured). MCB scheduled services will be started on PID 1 unless changed (see the BATCHPID SGS in Section 3.4.1, "Permanent SGSs", of the Q-LINK Installation Guide).

7.3.2.4 TIP Server Verification

If you cannot get your servers to run as a TIP transaction, try manually starting a BATCH mode server. The cause of the problem may be more easily found and corrected this way. Once the server can process requests from BIS while running in BATCH mode, try running it as a transaction again. You can also enter the transaction code for the Q-LINK server from any transaction terminal and you should receive a Q-LINK message on the system console noting that an invalid input was received, provided your VALTAB and transaction absolute are reasonably correct.

7.3.3 Starting Servers

After Q-LINK has been initialized by QMON, Q-LINK servers may be automatically started by QMON based on dynamic runtime configuration parameters. The starting of servers is based on the "INITIAL SERVERS IN CLASS" and "MAX SERVERS IN CLASS" runtime parameters (see the Server Class configuration parameters on page 3-29. Anytime a request is received from BIS and all active servers for the specified class are busy, and the maximum servers for the class are not yet active, Q-LINK will automatically start another server. Depending on the "MAX IDLE TIME" parameter, servers will stay available for a period of time. If no activity is received, within the idle time duration, the servers will automatically shut down (terminate). If idle time duration is set to zero (0), servers will not shut down until terminated via QMON. Setting idle time duration to zero may be desirable in some environments.

Q-LINK runstreams are optionally installed by COMUS (for a MODE=QLINK INSTALL):

SYSLIB*RUN$.QLPRD – A production mode server runstream.
SYSLIB*RUN$.QLTST – A development mode server runstream.
SYSLIB*RUN$.QLOFF – A production mode server for off-line processing.
SYSLIB*RUN$.QLMON – A QMON run for console communication.

For a MODE=QLINKx (wherex=A through K) INSTALL:

SYSLIB*RUN$.QxPRD – A production mode server runstream.
SYSLIB*RUN$.QxTST – A development mode server runstream.
SYSLIB*RUN$.QxOFF – A production mode server for off-line processing.
SYSLIB*RUN$.QxMON – A QMON run for console communication.

These runstreams are installed for the three batch server classes (PROD1, TEST1 and OFFLIN) configured at KMSYS Worldwide as the default classes on the release tape. The fourth runstream is for the QMON operation interface.

If you wish to start a TIP server as BATCH, you may copy one of these server runstreams and alter the copy accordingly. The same Q-LINK absolute can run as either a batch program, a TIP transaction or a TIP on-line batch transaction. The server type specified via the dynamic runtime parameter, "COMPOOL, BATCH OR APPL GROUP NBR", only controls how Q-LINK auto-starts its servers. At any time, you can start a batch type server manually, even if server type is TIP.

For TIP servers that use MCB instead of COMPOOL (the server class parameter, "COMPOOL, BATCH OR APPL GROUP NBR", is set to an application group number), a background QMON execution for that application group ***must*** always be active for Q-LINK to automatically start the servers. This is due to the addressing constraints imposed upon Q-LINK by the MCB; i.e., the Q-LINK CDB cannot access MCB directly. To create a QMON background runstream, make a copy of the QMON runstream and add the "S" option to the QMON processor call. We recommend the S-option that will inhibit the QMON privileged functions for the background run. The "L", "U" and "R" options should not be used. The copied runstream may be started by the same means the CDB is initialized for TIP servers (see Section 7.3.2.2, "TIP Servers" on page 7-4).

7.3.4 Servers Accessing TIP Data through UDS Control

When a Q-LINK server processes a request that will access TIP data under UDS Control, the server must, at some point, make its connection to TIP. If the server is a transaction, the connection will take place during scheduling of the server, not during the execution of the request. For batch servers, the connection will be made implicitly upon first TIP I/O unless the Q-LINK request explicitly connects via the CONNECT command; however, for an implicit connect, code in UDS control waits until a contingency 350012 (attempting a TIP executive request without a SLOP table) occurs before making the actual connection. In addition, the EXEC only allows 500 contingencies for any executing program before the program is terminated. This contingency processing implies that a batch server that stays resident on the system (in an idle state when not processing a request), will terminate after 500 requests. To avoid getting this contingency, require Q-LINK requests to use the CONNECT command to explicitly connect to TIP.

7.4 The QMON Operations Interface

QMON provides two important services. First, when started, it reads the dynamic configuration element and, if necessary, initializes the Q-LINK common data bank. Secondly, QMON provides the operations staff and the system administrator a method of monitoring Q-LINK. QMON may be operated from a demand terminal as a demand run, or from the system console as a batch run. Using QMON, Q-LINK server status may be displayed, servers may be locked, unlocked or shutdown. QMON also can be used to dump the Q-LINK common bank.

7.4.1 Starting QMON

The first step in activating Q-LINK is to start QMON to initialize the Q-LINK common bank based on the runtime configuration. Once QMON has initialized the common bank, it is no longer needed for normal Q-LINK operation except for TIP servers using the MCB instead of COMPOOL (see Section 7.3.3, "Starting Q-LINK Servers", page 7-5).

QMON accepts an integer in the first parameter field of the processor call. This parameter field is used to input a number (1 through 9) which represents the application group for which this execution of QMON is to start TIP servers automatically utilizing MCB instead of COMPOOL. If no application group is specified, QMON will default to the lowest numbered application group defined in the Q-LINK configuration. The format is as follows:

@QMON[x,*options*] *application-group-number*

Where x represents the mode used on the COMUS INSTALL of Q-LINK (QMON for the default mode and QMONA through QMONK for modes QLINKA through QLINKK).

QMON execution options are:

- C Used to allow QMON to be executed from a batch runstream, receive its input commands from the same batch runstream, yet display its output to the system console.
- I Used with the U-option so that only the classes named on data cards following the @QMON will be updated.
- L Ensure the CDB is loaded (L) and exit.
- R Used with the L-option to force a RESET command after ensuring the CDB is loaded and configured.
- S Inhibit QMON privileged registration with the CDB. Since only one user at a time can be registered to perform QMON privileged functions, this option allows multiple QMON runs to perform concurrent non-privileged functions such as status keyins.
- U Used with the L-option to update the configuration information in the CDB even if the CDB was previously loaded.
- Z Force the QMON idle activity to raise to real-time status. This will prevent the Q-LINK CDB from being swapped out of memory by the EXEC when Q-LINK is idle.

QMON may be executed at any time in order to monitor or communicate with the Q-LINK. All QMON key-ins are entered from either a demand terminal or the system operator console using the following format:

From the system console:

II *run-id command*

From a demand terminal:

▶ COMMAND ▶*command*

The following sections describe the QMON commands. Commands are the same for both demand terminal and system console input.

7.4.2 QMON Commands

Command abbreviations, where allowed, are shown in parentheses ( ):

ABORT

The ABORT command will immediately abort QMON. Normally, QMON is shutdown using the TERM or QUIT command. Using the ABORT command to terminate QMON will not cause server programs to abort.

CONFIG (C)

The CONFIG command will display information for servers configured. Configuration information includes:

- Server class name (Class Name)
- Server mode (MD / P=Production, D=Development)
- Server type (TY / C=COMPOOL, B=Batch, 1 .. 9=MCB)
- And logging on or off (LG / Y=Yes, N=No)
- Maximum servers in class (Active Mx)
- The number of servers initially started (Active Itl)
- Maximum number of output or reply lines (Max Lines)
- Maximum time for request (Max Run)
- Idle time before server terminates (Max Idle)
- Maximum number on queue (Max Que)
- Warning level on queue (Que Wrn)
- VALTAB transaction code for TIP servers (Valtab TCode).

If the server is an auto–starting TIP server, the VALTAB transaction code will be displayed. If the server is an auto–starting batch server, this field will be blank. If the field contains "*NONE*", the server is not auto–starting.

The following is an example of the QMON response to the CONFIG command:

```
»Max servers: 5  Insfile: SYS$LIB$*QLINKK
Class  MTL Active Max    Max      Max   Max Que Valtab
Name   DYG Mx/Itl Lines  Run      Idle  Que Wrn TCode
------ --- ------ ------ -------- ----- --- --- ------
PROD1  P1Y  4/ 2    2500 00:50:00 01:00   5   3 QLKBKA
DEV    DBN  2/ 0    2000 00:05:00 00:04   5   3 *None*
OFFLIN P1Y  3/ 1    1000 00:03:00 00:05   3   0 QLKBKC
TEST1  DBN  1/ 1    1000 00:03:00 00:05   3   0
SYSTST DCN  3/ 0    7500 00:10:00 00:05   5   0 QLKBKE
DEFALT PCN  3/ 0    7500 00:01:00 00:05   5   0 QLKBKE
```

DUMP (D)

DUMP will snap the current state of the Q-LINK common bank to a temporary print file (if DEMAND) or the run's PRINT$ file (if BATCH). The PRINT$ file is BRKPTed following the DUMP command.

EX *server-run-id*

The EX command is used for the orderly shutdown of individual Q-LINK servers programs. The *server-run-id* can be obtained with the active server status command (T) or the full status command (S).

HELP (?)

List QMON commands and functions. This command may also be issued by using the question mark (?).

The following is the display of the information provided when issuing the HELP command:

```
QMON Commands:
  ABORT  - Aborts the QMON program
  Config -Displays the configuration
  Dump   - Dumps Q-LINK CDB
  MI     - Miscellaneous information
  Quit   - Quits the QMON program
  Reset  - Resets counters and flags in Q-LINK CDB
  SB     - Displays Q-LINK statistics
  SS     - Q-LINK status
```

```
  Status - SS and T combined
  T      - Active servers
  End-of-File - same as Quit
Privileged QMON Commands:
  EX serverid        - Exits serverid
  INIT               - Inits Q-LINK CDB
  LOCK [class|ALL]   - Locks class or all classes
  TERM               - Terminates QMON and servers
  UNLOCK [class|ALL] - Unlocks class or all classes
```

All commands can be abbreviated to the uppercase portion of the command shown above (e.g., S for Status).

INIT

The INIT command will cause an immediate reload of the Q-LINK common bank. QMON will not accept this command if any servers are active. This command is equivalent to the EXEC RL-keyin using the E-option (error).

LOCK {*class-name* | ALL}

LOCK is used to suspend Q-LINK requests for the given server class, or for classes. Requests from BIS to a locked Q-LINK class will cause a return to the error label (with appropriate error status set) in the BIS run.

ALL will lock all Q-LINK classes and prevent the scheduling of any Q-LINK requests.

Entering a server class name will lock all servers in the specified class.

When a server class is locked, either individually or via the ALL option, requests in process will continue. Any queued or incoming requests will be returned with an error status.

MI

The miscellaneous information command provides release level identification information for each of the components of Q-LINK.

The first line identifies the date and the time the common data bank was initialized.

"CDB Level" displays the generation id from the COMUS BUILD of Q-LINK and the KMSYS Worldwide release identification level from the release tape.

"QMON Initialization" shows the date and time that QMON was called by the privileged caller.

"Privileged QMON user" shows the run-id of the user currently executing QMON in privileged mode. If an asterisk (*) precedes the run-id, the caller is the privileged user.

"Application group ... " shows the configured application groups and entry points.

The remainder of the display shows the software release levels of each component of Q-LINK currently being accessed by each active server.

The example, below, shows server class "PROD1" has one server currently linked to BIS and the Q-LINK external run function (QLKFUN) while the other server for PROD1 is not currently busy. Likewise, the one active server for server class DEV is not busy. Server class OFFLIN is busy processing requests from QLSIM and is not linked to BIS; therefore, the BIS level is flagged as "*Unknown*".

```
CDB Initialization: 09/03/99 15:49:14
CDB Level: Q-LINK-6R6 (Release 6R6)
```

```
QMON Initialization: 09/03/99 18:50:54
Privileged QMON user: *SAM
Application group: 1 Name: APPONE MCB entrypoint: 400612001000
Application group: 2 Name: APPTWO MCB entrypoint: 405055001000
Application group: 7 Name: APPSVN MCB entrypoint: 000000000000
[--Server---] Server/BIS/QLKFUN
Class Run Id [------Levels------]
------ ------ ----------------------
DEV QLDEV 6R1
PROD1 *00B5Q 6R1
PROD1 *00B5S 6R1/35R1/6R1
OFFLIN *00B5Z 6R1/*Unknown*/6R1
OFFLIN *00B63 6R1/*Unknown*/6R1
OFFLIN *00B6V 6R1/*Unknown*/6R1
```

QUIT (Q)

Terminates QMON without affecting servers. This is the normal way to exit QMON.

RESET (R)

This command resets the pending server start count for all classes to zero and then attempts sufficient starts to bring all classes to the initial active count that was configured.

STATUS (S)

The STATUS command will display status information for all active servers. The output from the STATUS command is equivalent to the SS command followed by the T command shown below.

SB

The statistics (SB) command provides information about active/busy servers and the activity in the transfer buffer of the common data bank.

The first line displays the total number of servers currently active and the number of servers currently busy.

The remainder of the display shows statistical information on the transfer buffers internal to the common bank. These buffers are used to transfer data between the server and BIS or QLSIM.

"Buffer count" is the number of buffers generated in the common bank.

"Requests Total" is the number of buffer requests made. "With 0 queued" is the number of buffer requests made without any queued buffer requests.

"Requests Requeued" is the number of buffer requests that have come off of the queue and been subsequently requeued. If this number is high, contact KMSYS Worldwide for assistance.

The following is an example of the SB command:

```
Servers Active/Busy: 5/3
        Transfer Buffer Statistics
Buffer count: 3
Requests Total/With 0 queued: 1932/1932
Requests requeued: 0
Requests queued at depth n
  1:            0           0           0           0
```

SS

The class status command provides statistics about all classes configured. Status information for each class includes the following:

- Class name (Class Name)
- If the server class was locked with a global lock; i.e., "LOCK ALL" (GL / Y=Yes, N=No)
- If the server class was locked with a class lock; i.e., "LOCK class-name" (LK /
- Y=YES, N=No)
- Total number of servers currently active (Active Cur)
- Maximum active servers allowed for the class (Active Max)

When the number of servers currently active for a class plus the number currently pending becomes equal to the maximum number of servers allowed for that class (as configured), an asterisk (*) is placed to the left of the Active column. When this happens, additional requests will be queued. The asterisk convention is also used when the number of requests currently queued for a class becomes equal to the maximum request queue size (also configurable). See "Queued Cur/Total" below.

- How many auto-start requests are pending (Pending Cur)
- How many auto-start requests via MCB are awaiting QMON action (Pending
- MCB)
- How many auto-start requests have failed (Strt Fail)
- How many user requests are currently queued (Queued Cur)
- How many user requests have been queued (Queued Total)
- How many user requests have been processed (Total Req's).

The following is an example of the QMON response to the SS command:

```
Class  GL  Active  Pending Strt  Queued     Total
Name   LK  Cur/Max Cur/MCB Fail  Cur/Total  Req's
------ -- -------- ------- ---- ---------- ------
PROD1  NN    2/  4   0/  0    0    0/    23   4212
DEV    NN    1/  2   0/  0    0    0/     1    456
OFFLIN NN *  3/  3   0/  0    0    1/   104    201
TEST1  NN    0/  3   0/  0    0    0/     1     72
SYSTST NN    0/  5   0/  0    0    0/     0      0
DEFALT NN    0/  5   0/  0    0    0/     0      5
```

In this example, there are six active servers; two in class PROD1, one in class DEV and three in class OFFLIN.

The OFFLIN server class only allows three active servers at any given time and three are currently active. And, since ACTIVE CUR + PENDING CUR = ACTIVE MAX, a fourth request for the OFFLIN server class has been queued and the status entry for this class marked with an asterisk (*).

In the entire PROD1 class, 4212 requests have been processed, 23 requests have been queued, and none are currently queued. A queued request is processed when a server in the specified class becomes idle, or when a new server in the class is started.

The last three registered server classes, TEST1, SYSTST and DEFALT, are not currently active.

T

The active server status command provides statistics about all servers currently active. Status information for each server includes the following:

- Class the server is running under (Server Class)
- Server run-id (Server Run Id)
- Server status (Server St / Av=Available, Bz=Busy, Wt=Waiting)
- Number of lines currently returned to the BIS run or QLSIM batch run (Reply Lines)
- Run-id of the requesting BIS run or QLSIM batch run (Requester Run Id)
- Station-id requesting service (UsrStn)
- BIS user-id of the requester (Requester User Id)
- BIS department number of the requester (Requester Dept)
- The number of server requests processed since the server was started (Server UseCnt).

The following is an example of the QMON response to the T command:

```
[----Server----] Reply [--Requester (BIS/QLSIM)---] Server
Class Run Id St Lines Run Id UsrStn User Id Dept UseCnt
------ ------ -- ----- ------ ------ ------------ ---- ------
PROD1 *00B5Q Av 3011
PROD1 *00B5S Bz 450 BIS 920 CLAUDIA 4 321
DEV QLDEV Av 358
OFFLIN *00B5Z Bz 3000 INFOQD 624 LINDA 4 201
OFFLIN *00B63 Wt 2350 INFOQD 949 DEVON 4 46
OFFLIN *00B6V Bz 1700 INFOQD 756 MARIE 4 31
```

This example shows the same active servers as in the example for the SS command above.

Individual servers within a class are identified by their run-id (Server Run Id). Server *00B5Q is free (idle) and has processed 3011 requests from BIS since it was started.

Server *00B5S is currently active processing a request from BIS user-id CLAUDIA at station 920. The request has returned 450 lines to the BIS run that issued the @QLK external run function call.

Note that the server run-ids also show that the servers in class PROD1 and OFFLIN are TIP transactions, while the server DEV is batch.

While DEV is idle, OFFLIN is busy with requests from stations 634 and 756, and is in a wait state on the request from station 949.

TERM

The TERM command is used to shutdown Q-LINK, including all currently active Q-LINK servers. TERM will shutdown all idle servers, and terminate QMON if all servers have been shutdown. Servers which are active when the TERM command is entered are not shutdown. The QMON run will terminate only after all server modules have shutdown.

UNLOCK {*class-name* | ALL

The UNLOCK command will reverse a previously entered LOCK command for a specific server class, or for all classes. After the UNLOCK, the server class is once again eligible to have Q-LINK requests routed to it for processing.

ALL will unlock all Q-LINK classes and resume the scheduling of Q-LINK requests.

Entering a server class name will unlock all servers in the specified class.

?

The question mark (?) is equivalent to the HELP command.

7.4.3 Reinitializing the CDB Security

It is possible to lock yourself out of Q-LINK by having only invalid or unusable user-ids in the security SGS list. Since QMON requires access to the common bank to add a user-id to the list, if you cannot access QMON then you cannot add a valid user-id to the list. You can fix this problem by reloading the Q-LINK common data bank with an RL-keyin from the system console (using the name you choose for that bank at installation time) and then immediately performing a COMUS PROCESS command with a valid configuration.

Chapter 8: Q-LINK DDP Operations

Assuming that a valid DDP-PPC connection exists between systems, the following steps must be taken to allow Q-LINK to submit and process requests on multiple systems.

8.1 Step 1: Initiate the QLDDP Processor

Create a runstream that will allow DDP-PPC to initiate the QLDDP processor. A sample runstream is shown below. In this example, the Q-LINK software was installed using a "MODE=K" install in COMUS. Because of the "MODE=K" install, the QLDDP absolute name was changed to QLKDDP, and the Q-LINK product file to SYSLIB*QLINKK. The QLKDDP program is copied to the TPF$ file for execution. The absolute QLNK$CFIG, which contains configuration information required by the QLDDP program, must also be copied to the TPF$ file. The name of the QLNK$CFIG absolute does NOT change depending on the install mode. The runstream continues by freeing the Q-LINK product file (SYSLIB*QLINKK). The QLDDP program is executed. The commands that follow the execution determine if an error has occurred and control the disposition of the print file.

This runstream must be saved in the file DDP*RUN-STREAMS. The DDP-PPC facility schedules all runs from the DDP*RUN-STREAMS file.

```
@RUN QLKDDP,KMS-DDP/DDP,KMS,,9999/9999
@SETC,N .  Inhibit PAGE_EJECT
@FREE TPF$.
@ASG,T TPF$.,///5000
@ASG,A SYS$LIB$*QLINKK.
@COPY,A SYS$LIB$*QLINKK.QLKDDP,TPF$.
@COPY,AV SYS$LIB$*QLINKK.QLNK$CFIG/************,TPF$.
@FREE SYS$LIB$*QLINK.
@TPF$.QLKDDP .  Run the QLDDP program
@TEST TNE/0/S2 .  Did this run error ?
@SYM,D PRINT$ .  NO, delete the PRINT$ file
@FIN .  YES, print the PRINT$ file
```

8.2 Step 2: Execute CSUPDT

Assuming that the above runstream has been saved as an element named QLKDDP in the file DDP*RUN STREAMS, the next step is to make the DDP-PPC facility aware of the runstream. The process of making DDP-PPC aware of the QLKDDP runstream is accomplished with the DDP-PPC utility CSUPDT. The runstream shown below will register the QLKDDP runstream with the DDP-PPC facility.

```
@sys$lib$*ddp-ppc.csupdt,l
add program
name=qlkddp
type=batch
```

```
file=ddp*run-streams.qlkddp
account-number=kms-ddp
copies=single;
@EOF
```

The use of the CSUPDT processor is fully covered in the Unisys publication entitled Distributed Data Processing Implementation and Administration Guide (3787 3270 000).

8.3 Step 3: Configure Server Classes

When using the DDP-PPC facility with Q-LINK, corresponding server classes must be established on each host. For example, if the remote host is using a server class called PROD1, the same server class must exist on the local host. The server classes are established through the dynamic runtime configuration screens (see Section 3.6.1, "Server Dynamic Parameters for Classes").

All QLDDP error codes and their explanations may be found in Chapter 13, "Q-LINK DDP Errors"

Chapter 9: ACOB Library Considerations

There are several possible permutations of ACOB installations. These permutations may cause some problems when generating Q-LINK. The default generation Q-LINK expects the ACOB library and the CBEP$$ACOB module to be in the file SYSLIB*ACOB and the ACOB processor call to be @ACOB.

The ACOB library consists of one to three cataloged files (as installed by COMUS). The default library file is SYSLIB*ACOB. It will contain one (and only one) CBEP$$ACOB element. Depending on your choices during the ACOB installation, this element will have one of the following version names:

PART-LIB-CB	Utilized for collections not linking to the C$DML common bank but which use common banked versions of the other ACOB library routines.
FULL-LIB-CB	Utilized for collections linking to the C$DML common bank.
NO-LIB-CB	Utilized for collections not linking to ANY ACOB common bank.

Note: Refer to the "ASCII COBOL Compiler (ACOB)" section of the OS 2200 Software Products Installation Guide, 7831 0612, for a list of possible locations of the CBEP$$ACOB elements.

The other ACOB library files, which may optionally be created during the ACOB installation process, are:

SYSLIB*ACOB-DML	This file is created to hold the FULL-LIB-CB version of the CBEP$$ACOB element. It allows using the C$DML common bank even though that is not the default for your site. The ACOB documentation refers to SYSLIB*ACOB-DML as the "utility CBEP$$ACOB" file.
SYSLIB*ACOB-CB	This file is created to hold the PART-LIB-CB version of the CBEP$$ACOB element. It allows using the ACOB common banked libraries (except C$DML) even though that is not the default for your site. The ACOB documentation refers to this file as the "common bank CBEP$$ACOB" file.

Normally, you would generate Q-LINK to use the most common banked form of the ACOB libraries available on your system. If you have the C$DML common bank available, you should set the C$DML configuration parameters for the Q-LINK generation and ensure that the element, CBEP$$ACOB/FULL-LIB-CB, is included in the Q-LINK collections. If the FULL-LIB-CB is not the default CBEP$$ACOB on your system, you will need to use the INCLUDE SGS discussed below.

If your system has a pre-6R1 level of ACOB installed, you must either upgrade to a more current ACOB level, or compile and collect Q-LINK with an alternate ACOB processor and library. In this case, you MUST ensure the following:

- CBEP$$ACOB/NO-LIB-CB element is included in the Q-LINK collections;
- Full relocatable ACOB library file is LIBed in the Q-LINK collections;
- Non-reentrant ACOB compiler is utilized for the Q-LINK compilations (ACOB/BANKED on the release tape but normally loaded into a sideline file as ACOB).

Thus, use of an alternate ACOB level will require the CO$ACOBPROCESSOR SGS as discussed below, and may require the INCLUDE and/or NOT SGSs.

If you are installing Q-LINK using the C$DML common bank and your ACOB installation does not assume C$DML usage, you must explicitly include the CBEP$$ACOB/FULL-LIB-CB element. The following SGS will be correct (assuming standard library names):

```
INCLUDE SYS$LIB$*ACOB-DML,CBEP$$ACOB/FULL-LIB-CB
```

If you do not wish to use the C$DML common bank and your ACOB installation uses it by default, you must select the element, CBEP$$ACOB/PART-LIB-CB. In addition, if your ACOB library file does not contain the required relocatable versions of the ACOB library routines (one ACOB installation option is to delete those that are also contained in the installed common banks) you will have to manually load those routines from the ACOB release tape and place them in a sideline file. The following SGS will normally be correct for selecting the proper CBEP$$ACOB element. Also, see the CO$ACOBPROCESSOR SGS.

```
INCLUDE SYS$LIB$*ACOB-CB,CBEP$$ACOB/PART-LIB-CB
```

If you do not wish to use any ACOB common banked library routines, you must either include the CBEP$$ACOB/NO-LIB-CB element or prevent the inclusion of any CBEP$$ACOB element. Excluding ACOB common banked library routines can be performed with the following SGS:

```
NOT CBEP$$ACOB
```

If you have either your ACOB library or ACOB compiler installed in a non-standard file, you must also include a CO$ACOBPROCESSOR SGS (see the COMUS User Guide for additional information). This SGS is specified as:

```
CO$ACOBPROCESSOR CALL NAME IS ''q*f.ACOB'' ;
   OPTIONS ARE ces LIBRARY FILE IS ''q*f.''
```

The call name of the compiler and the library file are entirely arbitrary and must simply match the way ACOB is loaded on your system. The compiler options must include options for listing control. Options such as "T" (reverse DISPLAY and DISPLAY-1 usage defaults) will prevent proper operation of Q-LINK. If you need to specify ACOB "extra-options," you may supply them as an optional second subfield following the primary options. You may use the literal string, "NONE", for the options if you do not wish to specify any options.

Note: If you are unsure how your ACOB library is installed (i.e., what libraries, if any, are usable from common banks), you should look at the version names of the CBEP$$ACOB elements in the three library files mentioned in the above discussion. If you find a particular version and INCLUDE it, you will be using that form of the libraries. This procedure may be complicated by the fact that for compatibility with pre-COMUS ACOB installation procedures, many sites may not use the default ACOB library file names (these can be changed by using the INSFILEn parameter on the COMUS INSTALL ACOB command). You can determine which library files were registered as part of the ACOB installation by using any text editor to view the element:

```
SYS$*DATA$.CO$INSTALL$/COMUS$
```

If your site has installed ACOB in a nonstandard fashion or is running non-supported levels of software, you will have to determine the required library names, etc., manually.

Chapter 10: MCB Considerations

Since the TIP primitives that are used by Q-LINK have been replaced by KMSYS Worldwide code that can access COMPOOL or any configured MCB, there is no longer any default TIP library file. The KMSYS Worldwide replacement primitives maintain the Unisys name and calling conventions so that a site can replace them with their own customized versions, should it be necessary. The replacement would be done as in the past by supplying a CO$TIPPROCESSOR card with a library name. No CO$TIPPROCESSOR SGS or one without a library clause will default to the KMSYS Worldwide primitives.

If you desire to use the new capability to access COMPOOL and MCB from one absolute, make sure that you do not have a library field on any CO$TIPPROCESSOR permanent SGS. This permanent SGS may have been entered when the Q-LINK products defaults were defined during a previous COMUS BUILD session. You may view and/or reenter any permanent SGSs entered in this manner by using the "Q" option on the COMUS BUILD. Another place that permanent SGSs may reside is in the "MISC PARAMETERS" element which may have been named on the ENVIRON screen during the COMUS CONFIGURE session.

Even though one Q-LINK absolute can access any application group, users must be careful not to attempt to access more than one application group per server execution. The most likely incident is when a server is scheduled through MCB and a subsequent IMPART is done to a schema that requires a different application group. In this case UDS/DMS will return an error to the effect that the specified schema cannot be found (DMS ERROR-NUM=0166). This error is returned because UDS DMS will only look for schemas that belong to the application group that the request is currently associated with.

In order to avoid this error, the person responsible for configuring server classes should set up separate classes for each application group to be accessed.

An INVOKE or explicit CONNECT will maintain any previous application group association that may have been present. If there is currently no application group association, the connection will be through COMPOOL, without an application group, unless it is a user-explicit CONNECT with an application group specified.

To generate Q-LINK without requiring any TIP library, use the following SGS in the generation:

```
CO$TIPPROCESSOR CALL NAME IS NOT
```

A Q-LINK generated in this manner cannot invoke TIP schemas, use TIP servers or process TIP or TIPDMS DIO files, but is a fully functional Q-LINK in all other respects.

Chapter 11: QLK External Function Errors

The following is a list of error codes, explanations of errors and possible error resolutions for all errors reported to BIS runs by the @QLK external function and the QLSIM (Q-LINK/InfoQuest only) program:

ERROR CODE	DESCRIPTION
1	SPECIFIED SERVER NOT ACTIVE Explanation: The server class passed via the QLK function or the third specification field on the QLKSIM processor does not have any servers active and no servers were successfully activated by the server auto-start process (if configured). Resolution: If not using the server auto-start, you might configure it. If batch servers are taking too long to start, consider using TIP servers. If you do not use the server auto-start, consider having an initial number of servers started when the system is initialized by QMON (R-option in conjunction with the L-option or the RESET command), and carefully consider the effects of using the terminate idle server timer configuration. Investigate why auto-started servers are not able to start. Check the QMON STATUS key-in to verify that the requested class is in the CDB tables. If the PEND or FAIL or QMON columns of the class status are non-zero, investigate why auto-start requests are being delayed or rejected. Remember if using the QMON auto-start configuration (versus BATCH or TIP) that auto-start requests will be held by the CDB until a QMON or a QMON,S run is activated.
2	CDB ID CONSTANT VALIDATION FAILED Explanation: A BIS caller could not access all of the CDB due to an addressing error or the Q-LINK CDB has been corrupted. Resolution: Check your BIS main buffer pool address limits versus your CDB configuration to determine if an address conflict should occur. If the problem persists see the action for error 1.
3	*** Error code not currently in use. *** Explanation: Resolution:

ERROR CODE	DESCRIPTION
4	MAPPER RELEASED BY QMON Explanation: A BIS run which was waiting for a server to act was released because the server aborted processing the request. Resolution: Check for server aborts due to either E-keyins or other operational events. Report unexpected server aborts via the procedure outlined for common bank error 1.
5	MAX REQUESTS QUEUED FOR SERVER CLASS Explanation: More than the configured maximum number of requests for this server class would have been outstanding had this request been queued. Resolution: Investigate why requests are not completing faster. Consider increasing the number of servers that can be active. The auto-start and auto-terminate configuration can make this painless, and minimize the impact on operations at other times of the day. Ensure that the number of servers that can be auto-started is actually starting promptly and successfully. Consider moving long running requests from this class into another class to allow short requests to complete. Consider the priority and execution type of the server. TIP requests generally (based on VALTAB) have higher priority than BATCH requests and TIP transaction type servers would usually be configured for higher priority than TIP batch-transaction type servers. Check other system bottlenecks for resource problems including CPU/memory/IO saturation and DMS run-unit queuing problems.
6	SERVER REQUESTED ABORT OF REQUEST Explanation: A runtime error in the Q-LINK program was detected by the server (other than max reply lines, max time or a security error). Resolution: Retrieve the server log (if not already returned for DEVELOPMENT mode servers) and inspect it for error messages defining the type and cause of the error.
7	SERVER CLASS QUEUE NOT IN MCT Explanation: The server class passed via the QLK function or the third specification field on the QLKSIM processor is not currently configured in the CDB. Resolution: Check the spelling and configuration of the specified class name. Use the QMON CONFIG command to verify that the proper configuration is resident in the CDB.
8	USER ABORTED FUNCTION WITH MESSAGE WAIT KEY Explanation: The Q-LINK server has detected that the user has depressed the message wait key (or the F4 key when running through MCB) and has terminated processing of this request. When running through the QLSIM program, the @@X C and the operator II-key-in will also terminate a request in this fashion. Resolution: None.
9	ALL SERVER CLASSES LOCKED Explanation: The QMON program has been used to lock all request classes and prevent the queuing or acceptance of new requests. Resolution: Contact your system administrator to determine why Q-LINK access has been terminated.

ERROR CODE	DESCRIPTION
10	THIS SERVER CLASS IS LOCKED Explanation: The QMON program has been used to lock this specific class and prevent the queuing or acceptance of new requests. Resolution: Contact your system administrator to determine why Q-LINK access has been terminated for this specific class.
11	ASSIGNED SERVER VANISHED! Explanation: The assigned server was no longer in the CDB tables when BIS returned for/with the next block of lines. Resolution: See error code 1.
12	BAD RID - OPEN I/P TO READ Explanation: A nonexistent input RID was specified on the @QLK function or QLKSIM utility call line. Resolution: Verify the desired MODE/TYPE/RID combination for the request.
13	RID # TOO BIG FOR TYPE ON OPEN Explanation: The input RID number specified is larger than the largest configured in the BIS startup parameters. Resolution: Verify the desired MODE/TYPE/RID combination for the request.
14	LINE # TOO BIG ON OPEN I/P RID Explanation: The input RID did not contain even one line of data. Resolution: Verify the desired MODE/TYPE/RID combination for the request. Verify that the desired data is actually in the requested RID.
15	SYSTEM ERROR OPENING I/P RID Explanation: A BIS system error occurred trying to open the input RID. Resolution: Refer this problem to your BIS coordinator.
16	ERROR OPENING O/P RID Explanation: A BIS system error occurred trying to open the output RID. Resolution: Refer this problem to your BIS coordinator.
17	ERROR WRITING OUTPUT RID Explanation: A BIS system error occurred trying to write to the output RID. Resolution: Refer this problem to your BIS coordinator.
18	ERROR CLOSING OUTPUT RID Explanation: A BIS system error occurred trying to finish writing the output RID. Resolution: Refer this problem to your BIS coordinator.
19	ERROR WRITING DATE LINE Explanation: A BIS system error occurred trying to write the date line to the output RID. Resolution: Refer this problem to your BIS coordinator.

ERROR CODE	DESCRIPTION
20	ERROR WRITING HEADER LINE TO OUTPUT RID Explanation: A BIS system error occurred trying to write the header lines to the output RID. Resolution: Refer this problem to your BIS coordinator.
21	SYS MAX LINES REACHED Explanation: The Q-LINK user program was terminated when it tried to generate more lines of output than are permitted for this class. Resolution: Investigate how much data the request should be designed to return. Occasional large output can be allowed via the Q-LINK password on the @QLK function or QLSIM program call.
22	USER-SPECIFIED MAX LINES REACHED Explanation: The Q-LINK user program was terminated when it tried to generate more lines of output than were specified on the @QLK function or QLKSIM program call. Resolution: Investigate why the program attempted to return more data than the designer intended.
23	SYSTEM ERROR READING I/P RID Explanation: A BIS system error occurred trying to read from the input RID. Resolution: Refer this problem to your BIS coordinator.
24	CDB ADDRESSING WINDOW PROBLEMS Explanation: The combination of the BIS main buffer pool sizing and the CDB installation selected base address has produced an overlap of the CDB address space and the BIS buffer pool address space. Resolution: Either BIS must be reconfigured, or the CDB must be reinstalled in such a way as to eliminate the addressing overlap. See also the action for common bank errors 11 and 12.
25	UNABLE TO TRMRG$ MAPPER ACTIVITY Explanation: The ER TRMRG$ performed by the CDB when registering a BIS activity has been rejected by the EXEC. Resolution: If the problem persists, submit an EXEC dump immediately following the error occurrence, along with the information requested for an internal Q-LINK error, to KMSYS Worldwide, Inc., for analysis.
26	CDB FAILED INITIAL STATUS CHECKS Explanation: The CDB is improperly installed, or is from an incompatible level of Q-LINK. Or BIS was improperly generated with regard to the CDB bank name, type and/or BDI. Resolution: Investigate the type and status of the bank installed for the Q-LINK CDB and the configuration of the BIS TCF and the status and completion log of the BIS generation used to install the @QLK function. For BIS 34 and higher, there is no QLK CDB parameter card defining the CDB BDI. If the problem persists, submit an EXEC dump immediately following the error occurrence along with the information requested for an internal Q-LINK error to KMSYS Worldwide, Inc., for analysis.

ERROR CODE	DESCRIPTION
27	ERROR OPENING #INSERT RID Explanation: The Q-LINK function in BIS was unable to open the RID specified on the #insert directive. Resolution: Verify the desired MODE/TYPE/RID combination for the request.
28	ERROR READING #INSERT RID Explanation: An error occurred trying to read from the input RID. Resolution: Refer this problem to your BIS coordinator.
29	SYNTAX ERROR ON #INSERT DIRECTIVE Explanation: The format of the #insert directive is incorrect. Resolution: Verify the format of the #insert command from the Q-LINK documentation. Any comment on the directive must be preceded by at least one space, a period and at least one additional space, and the comment must follow all required fields.
30	SERVER CLASS MAXIMUM XQT TIME EXCEEDED Explanation: The Q-LINK user program was terminated when it tried to execute and consume more resources (SUPS) than are permitted for this class. Resolution: Investigate how long the request program should run. Consider processing fewer records during the request. Consider more efficient database location paths for the desired data. As a last resort, the execution time limit configured for the class can be increased, but this allows requests that might loop to consume unwarranted amounts of system resources. If requests requiring differing magnitudes of system resources are currently lumped together in one class, consider creating an additional class(es) each of which has an appropriate time limit for the type of processing required.
31	ACCESS NOT ALLOWED (#INSERT MODE EDITING) Explanation: The BIS run function attempted to access data within a MODE that is not allowed by the run registration RID for that run. Resolution: Verify the desired MODE/TYPE/RID combination for the #insert request. Request that your BIS coordinator update the run registration log for this run if you feel your need to access the data is legitimate. Alternatively, move the data to a MODE, which the run can access.
32	TOO MANY CHAINED #INSERTS ENCOUNTERED Explanation: When a #INSERT is encountered in a RID being read with #INSERT, the first RID is closed and the new RID is opened and read. Q-LINK does not keep a history of which RIDs have been processed in a chain of #INSERT commands, and so cannot explicitly detect a situation where a RID contains a #INSERT which directly or indirectly references itself. To prevent BIS from looping forever in such a case, Q-LINK counts the number of #INSERT directives encountered while reading from a RID selected by another #INSERT. If the count of chained #INSERT commands exceeds ten, this error is issued. Resolution: Check the #INSERT commands in the data stream, and ensure that the maximum chaining of RIDs is not greater than ten. Also, ensure that the chain of #INSERT commands never references a RID present earlier in the chain.

ERROR CODE	DESCRIPTION
33	SECURITY ERROR - NO ACCESS TO CLASS Explanation: Your BIS user group (configured in Q-LINK security) is not allowed access to the server class that you have selected. Resolution: Contact your site security officer to determine which server classes you can access.
34	SECURITY ERROR - COMPILE NOT ALLOWED Explanation: Your BIS user group (configured in Q-LINK security) is not allowed to compile requests with this server class. Resolution: Contact your site security officer to determine which server class you can use to compile requests.
35	SECURITY ERROR - RUN NOT ALLOWED Explanation: Your BIS user group (configured in Q-LINK security) is not allowed to run requests with this server class. Resolution: Contact your site security officer to determine which server classes you can use to run requests.
36	QLKFUN IMAGE LENGTH ERROR Explanation: A bad ACW was transferred to QLKFUN (BIS) from the server. Resolution: Contact KMSYS Worldwide, Inc.
51 through 81	COMMON DATA BANK DIRECTED ERRORS Resolution: For error codes 51 through 81, subtract 50 from the error code and use the result to reference the errors in Chapter 19, "Q-LINK Common Bank Error Conditions"; i.e., @QLK error code 68 can be found in the next chapter as error code 18 (68 -50 = 18), "INSUFFICIENT PRIVILEGE FOR REQUEST".

Chapter 12: Q-LINK Common Bank Errors

The following is a list of error codes, explanations of errors and possible error resolutions for all errors reported by QMON, Q-LINK servers and other Q-LINK utility programs:

ERROR CODE **DESCRIPTION**

1 SERVER TABLE NOT FOUND

Explanation: Internal error. A server run-id was not found in the CDB when it was expected to be there.

Resolution: Take a dump of the common bank immediately following the error using the QMON DUMP command. Take a dump of the utility program reporting the error using @PMD. Initialize the common data bank using the QMON INIT command and try again. Submit the dumps and a listing of your configuration (both generation and runtime) to KMSYS Worldwide, Inc., for analysis.

2 UNKNOWN FUNCTION REQUESTED

Explanation: Internal error. An unused or out of range function code was passed to QCOM (part of the Q-LINK server program) on a request call.

Resolution: See error code 1.

3 INIT REQ AND SERVER ALREADY ACTIVE

Explanation: Internal error. A server initialization function was received by the CDB from a run-id, which was already marked as active in the CDB tables.

Resolution: See error code 1.

4 MAXIMUM NUMBER OF SERVERS ACTIVE

Explanation: Configuration or operational error. The number of servers allowed to be active concurrently for all classes is too small for the number of servers attempting to connect to the CDB. The runtime configuration for maximum servers active needs to be as large as the sum of the maximum active servers for all classes.

Resolution: Change your runtime configuration to reduce the number of servers allowed for some classes, or increase the maximum number of servers globally allowed active by the CDB.

ERROR CODE	DESCRIPTION
5	QMON IS SHUTTING DOWN THIS SERVER -OR-SERVER WAS ABORTED (E-KEY, ...) -OR- Q-LINK TIMER ACTIVITY IS SHUTTING DOWN SERVER Explanation: This condition is expected by Q-LINK and, as such, it should not appear as an error; i.e., the CDB is requesting a server to exit. Resolution: Investigate QMON EX and TERM command usage at your site. Ensure that the runtime configuration for the time a server is to remain idle before exiting (DURATION SGS) is as large as you would like it to be.
6	MAXIMUM SERVERS ALREADY ACTIVE FOR CLASS Explanation: Configuration or operational error. The number of servers allowed to be active concurrently for this class is too small for the number of servers attempting to connect to the CDB. The only time this condition should occur is when someone is manually starting servers, or if auto-started servers were held for some reason while one or more QMON RESET or INIT commands were done. Resolution: If you really want more servers active, update the runtime configuration through COMUS. If not, investigate why servers are being delayed in starting, or who is manually starting extra servers.
7	NO CLASS QUEUE SLOT FOUND Explanation: A server attempted to initialize with a class name which was not in the CDB's current configuration. This could be caused by a misspelled server's class name on the @QLK command in a manually started batch server. This could also occur if the CDB was initialized with a configuration not containing this class name. Since class names cannot be dynamically added to the CDB, you can successfully process a configuration update but still not have access to the new server class. Resolution: Verify that the server name requested is really in the current COMUS configuration. Verify what classes are currently available in the CDB by using the QMON CONFIG command. If you need to add the class to the CDB, you must shut down the Q-LINK system using the following commands to QMON: LOCK, ALL, TERM and then initialize the Q-LINK system using the QMON INIT command.
8	ABORT OF A REQUEST SIGNALED REQUESTED FROM MAPPER -OR- MAPPER HAS ABORTED (E-KEY,...) Explanation: This error is not fatal! Someone has externally requested the termination of the current request for this server. Possible cause includes a shutdown of the BIS system that submitted this request to the server. Resolution: Investigate the BIS down/purge times as well as any manual shutdowns of BIS or any cases of BIS aborting.
9	**** Error code not currently in use. **** Explanation: Resolution:

ERROR CODE	DESCRIPTION
10	ERROR LOCATING OR READING CONFIG ELT Explanation: All the Q-LINK utility programs and Q-LINK batch servers locate the runtime configuration by finding an absolute element called QLNK$CFIG in the file from which they were executed. The TIP servers are passed the filename from which the CDB configuration was initialized, and they expect the QLNK$CFIG absolute to be in that file. BIS has the CDB type as specified by a parameter card and does not need access to the configuration absolute. If the configuration element cannot be found or is badly formatted, or if an I/O error occurs when attempting to read it, then this error is reported. This can be caused by a hardware or software damaged execution file, by copying a Q-LINK program into a different file (i.e., TPF$) and not copying the configuration element also, or by attempting to process a configuration element produced by a different level of Q-LINK. Note that immediately following a Q-LINK generation and/or install, the installation file (Q5 or SYSLIB*QLINKx) contains a dummy QLNK$CFIG element which cannot be loaded by the Q-LINK programs.
11	HIGHEST ADDRESS REQUIRED FOR CDB IS GREATER THAN 0777000 (1ST PCT BLK ADDR) Explanation: The runtime configuration of maximum servers active and number of classes defined, combined with the base address for the CDB chosen at install time, has produced a CDB whose highest address exceeds 0777000. Resolution: Reduce the upper address limit of the CDB by reducing one (or more) of the following: CDB base address (INSTALL), maximum servers concurrently active (CONFIGURE), or number of defined classes (CONFIGURE). Remember not to reduce the base address of the CDB below the highest address of the main buffer pool of BIS or intermittent addressing status errors may be returned to @QLK users. If you are running a very large configuration and running multiple BIS installations or InfoQuest batch servers, you can split these to use a second (or more) CDB although it seems extremely unlikely that anyone would have a configuration this large.
12	Q-LINK UNKNOWN ERROR Explanation: Resolution: Contact KMSYS Worldwide, Inc.
13	UNABLE TO DO AN ER TO TRMRG$ Explanation: The ER TRMRG$ performed by the CDB when registering a server or QMON run has been rejected by the EXEC. Resolution: If the problem persists, submit an EXEC dump immediately following the error occurrence, along with the information requested for an internal Q-LINK error, to KMSYS Worldwide, Inc., for analysis.
14	ALL CLASSES LOCKED/UNLOCKED Explanation: A QMON LOCK (UNLOCK), ALL directive was performed when the global request lock was already set (clear). Resolution: Perform a QMON STATUS key-in prior to issuing the LOCK/UNLOCK command. Investigate the usage of the QMON LOCK/UNLOCK command at your site.

ERROR CODE	DESCRIPTION
15	QUEUE CLASS NOT FOUND Explanation: A QMON LOCK (UNLOCK) class directive was performed for a class whose name is not currently in the configuration tables of the CDB. Alternatively, an attempt was made to update the configuration by adding a class name to the CDB dynamically. Yet another alternative, by some strange timing quirk, a class name was removed from the CDB (by a QMON INIT command) while a second QMON was auto-starting a TIP MCB connected server. Resolution: Verify the currently active configuration using the QMON CONFIG command. If you have added a class by the configure process, but the CDB was already initialized, you must use the QMON TERM and INIT commands to allow the reloading of the configuration data into the CDB.
16	SERVER INDEX OUT OF RANGE Explanation: An invalid server index was used on a QMON EX key-in. Resolution: See error code 1.
17	QMON ALREADY ACTIVE Explanation: Only one run can be registered for the QMON privileged commands at any one time. Resolution: Use the "S" option on one or all of the QMON executions. The QMON copy, that received this error, automatically acted as if the "S" option were specified. The QMON STATUS key-in will display the run-id and start time of the currently registered privileged QMON run.
18	INSUFFICIENT PRIVILEGE FOR REQUEST Explanation: Certain CDB requests require that you be signed on under one of a specific list of user-ids before you are allowed the privilege associated with that request. Resolution: Check the list of user-ids configured in the security SGSs for utility security. Check the user-id of the offending run to make sure that it is configured. If the user is a TIP transaction server, its pseudo user-id will consist of the program name prefixed with at '$' character; i.e., the QLINK transaction program requires configuration of a $QLINK user-id.
19	MUST HAVE QMON UNLIMITED PRIVILEGE FOR REQUEST Explanation: Certain CDB requests require that the caller be the privileged QMON user registered with the CDB. If a non-privileged user requests such a function (reload the CDB, etc.) it will be rejected. Resolution: The "S" option on QMON prevents registration as the privileged user and can lead to this error if privileged commands are then attempted. As pointed out for error 17, if a user is not the first to attempt to register for these privileges, it is as if he used the "S" option when starting QMON, and that can lead to this error. Check the QMON privileges for the user in the security SGSs.

ERROR CODE	DESCRIPTION
20	BANK NOT YET LOADED

Explanation: Only the QMON program can initialize the CDB following a COMUS installation, a system reboot or a console reload bank keyin. If any other caller attempts to link to the CDB prior to QMON beginning the bank initialization, the caller will be rejected.

Resolution: The QMON program must be executed in some fashion before starting servers or other utility programs. BIS can be started before the CDB is initialized, but cannot access Q-LINK until the CDB is initialized. An @SYSLIB*QLINKx.QMONx ,L in either the system startup runstream or the BIS startup runstream or the server startup runstream will allow the common bank to be loaded in an automatic fashion. Be sure to use the QMON and the QLNK$CFIG element from the current installation file as whatever configuration is in the execution file will be the one which is loaded.

21 TEMPLATE BANK ILL-FORMED -OR- REQUESTER NOT IN QUARTER-WORD MODE

Explanation: Either the CDB absolute has been damaged by some software or hardware malfunction, or the CDB does not match the level of the Q-LINK utility/server program being executed.

Resolution: Verify the level of Q-LINK used when installing the CDB and the level of Q-LINK being used for execution. If all else fails, de-install the Q-LINK software and then repeat the installation of the Q-LINK software with COMUS.

22 CDB CONTINGENCY (INCL RELOAD)

Explanation: Some contingency has occurred which the CDB is not prepared to handle. This will occur if someone reloads the CDB using the console key-in or a privileged reload program. Other errors constitute an internal error in the Q-LINK system. The contingency packet information is returned to the caller in A1 and A2.

Resolution: Investigate the usage of reload key-ins/programs affecting the Q-LINK CDB. Check for installation of other software using COMUS that uses the same BDI or bank name as the Q-LINK CDB. If the problem persists, see the action for error 1.

23 ILLEGAL CB FUNCTION

Explanation: Internal error. An unused or out of range function code was passed to the CDB in A0.

Resolution: See error code 1.

24 SERVER(S) ARE ACTIVE

Explanation: Attempt to terminate Q-LINK with the QMON TERM command while servers were processing a request.

Resolution: Use the QMON LOCK ALL command to prevent new requests from starting. Once all active requests have finished, retry the TERM command.

ERROR CODE	DESCRIPTION
25	A MAPPER RUN WAS RELEASED Explanation: Attempt to initialize the Q-LINK CDB with the QMON INIT command while a BIS run was waiting for a server activity to process. The BIS run has been released. Resolution: The INIT command should be retried after first using the QMON LOCK, ALL and allowing active requests to terminate.
26	NO BIS RUN TO RELEASE Explanation: A BIS run was to be released by the CDB and no BIS run was waiting for the specified server. Resolution: See error code 1.
27	SERVER NOT CLOSED; I.E., IS BUSY Explanation: The server specified on an EX/TERM key-in could not be shutdown since it was not idle. Resolution: Lock the class queue using the QMON LOCK command and then wait till the server is idle before terminating it.
28	SERVER NOT CLOSED; I.E., IS TERMINATED/ABORTED Explanation: The server specified on an EX/TERM key-in could not be shutdown since it is in a termination or abort state. Resolution: If this condition does not clear shortly, then this is an internal error and it should be reported as for error 1.
29	AN IDLE SERVER WAS RELEASED Explanation: This status is used between the CDB and QMON as part of the TERM key-in processing. A message is issued to the user and the request will be retried. Once all idle servers are terminated, QMON will finish. Resolution: None.
30	BAD BASE/RETURN ON LBJ Explanation: A caller to the MCB had invalid information captured by the hardware related to the BDR on which the CDB is based and the return address portion of the linkage register. Resolution: See error code 1 if this error persists.
31	USER BUFFER ACW INVALID Explanation: A caller to the CDB passed an invalid buffer length/address for a function requiring data transfer between the program's DBANK and the CDB data areas. Resolution: See error code 1.

Chapter 13: Q-LINK DDP Errors

The following is a list of error codes, explanations of errors and possible error resolutions for all errors reported by QLDDP:

ERROR CODE	DESCRIPTION
1200	DDP INIT. BIND Error. Class: ### Detail: ### Explanation: Q-LINK could not connect to the DDP-PPC facility. The Class and Detail status codes are listed in the Unisys manual OS2200 Distributed Data Processing (DDP-PPC/DDP-FJT) Messages Reference Manual (3787-3528).
1201	DDP INIT. OPEN Error. Class: ### Detail: ### Explanation: Q-LINK issued an OPEN request that could not be completed by the DDP-PPC facility. The Class and Detail status codes are listed in the Unisys manual OS2200 Distributed Data Processing (DDP-PPC/DDP-FJT) Messages Reference Manual (3787-3528).
1202	DDP INIT. No REPLY received. Explanation: This is a protocol error. The QLKDDP program did not reply correctly to the Q-LINK request to initiate a conversation. Contact KMSYS Worldwide.
1210	DDP XFER. Expected OUTPUT EOF. Explanation: This is a protocol error. Q-LINK was notified that no more data exists to be returned. Q-LINK is expecting a final status message to confirm the OUTPUT EOF status; but the received message was not the correct type. Contact KMSYS Worldwide.
1211	DDP RECV. RECEIVE Error. Class: ### Detail: ### Explanation: Q-LINK received an error from the DDP-PPC facility while attempting to receive data. The Class and Detail status codes are listed in the Unisys manual OS2200 Distributed Data Processing (DDP-PPC/DDP-FJT) Messages Reference Manual (3787-3528).
1212	DDP SEND. SEND Error. Class: ### Detail: ### Explanation: Q-LINK received an error from the DDP-PPC facility while attempting to send data. The Class and Detail status codes are listed in the Unisys manual OS2200 Distributed Data Processing (DDP-PPC/DDP-FJT) Messages Reference Manual (3787-3528).

ERROR CODE	DESCRIPTION
1213	DDP TOKEN. SEND TOKEN Error. Explanation: An error occurred while Q-LINK was attempting to pass the control token to the other member of the conversation. Contact KMSYS Worldwide.
1220	DDP EXIT. Receive CLOSE Error. Class: ### Detail: ### Explanation: Q-LINK received an error from the DDP-PPC facility while attempting to close the current conversation. The Class and Detail status codes are listed in the Unisys manual OS2200 Distributed Data Processing (DDP-PPC/DDP-FJT) Messages Reference Manual (3787-3528).
1221	DDP EXIT. Expected CLOSE message. Explanation: This is a protocol error. Q-LINK has received the final status message from remote, and was expecting a close message which was not received. Contact KMSYS Worldwide.
1222	DDP EXIT. Send CLOSE Response failed. Class: ### Detail: ### Explanation: Q-LINK received an error from the DDP-PPC facility while attempting to close the conversation. The Class and Detail status codes are listed in the Unisys manual OS2200 Distributed Data Processing (DDP-PPC/DDP-FJT) Messages Reference Manual (3787-3528).
1230	DDP-PPC Abort. Explanation: This error can be caused when one party of DDP-PPC conversation is abnormally terminated.
1231	DDP Facility not configured. Explanation: This message is issued when a request is made for remote execution of a Q LINK request and the DDP-PPC facility has not been configured or enabled. In order to enable the DDP-PPC facility, Q-LINK must be build with the required DDP-PPC files and the dynamic DDP switch must be set to ON. These fields are located on the COMUS configuration screen BUILD and BASIC1. Contact your Q-LINK coordinator.

Chapter 14: Pre-GENed Q-LINK Parameters

The following configuration screens show the default generation parameters. These were the parameters used to build the Q-LINK contained on the product release tape. They are not the defaults used when generating Q-LINK at your site. In addition, the parameters shown on the "INSTALL" screen should be examined carefully and changed where necessary, prior to installing the default generation as outlined in the Quick Installation Procedure in Chapter 1.

The following screen is the first screen for Selection 37 from the main menu. The parameter values shown were used to generate Q-LINK absolutes on the release tape.

```
Enter + or - to view continuations of this screen.
                                                          SCREEN: ENVIRON
 These parameters define information which can only be changed by a Q-LINK
 system generation. Enter a question mark (?) in any field and transmit for
 information on how to configure that parameter. Other generation parameters
 can be viewed with selections  from the main menu (use the BACK command).


 FILENAME OF MISC PARAMETERS ELT              QLINK*MAINT
 ELEMENT NAME OF MISC PARAMETERS              EXGENSGS
 OBJECT COMPUTER TYPE                         THIS SYSTEM
 USE COMMON BANKED SORT LIBRARY               FALSE
 ACOB LIBRARY CBEP$$ACOB FILE                 THIS SYSTEM
 ACOB C$DML BANK TYPE                         THIS SYSTEM
 ACOB C$DML BANK SIZE                         THIS SYSTEM
 EXECUTION COMMON IBANK NAME                  NONE
 EXECUTION COMMON IBANK BDI                   NONE
 EDITING COMMON IBANK NAME                    NONE
 EDITING COMMON IBANK BDI                     NONE
 DMS I/F COMMON BANK NAME                     NONE
 DMS I/F COMMON BANK BDI                      NONE
 FEATURE COMMON BANK NAME                     NONE
 FEATURE COMMON BANK BDI                      NONE
```

The following screen is the second screen for Selection 37 from the main menu. It can be reached by typing a plus sign (+) in the home position on the previous screen and transmitting. The parameter settings on this screen are required to install Q-LINK. If you are using the Quick Installation Procedure outlined in Chapter 1, make sure the common data bank BDI and bank name are not currently in use at your site. This can be accomplished by looking for them with an editor in the following element: SYS$*DATA$.CO$INSTALL$/COMUS$. In addition, the installation process needs to know the location of the BIS relocatable output (MRO) file on your system when installing Q-LINK to interface with BIS levels earlier than level 35.

```
Enter + or - to view continuations of this screen.
                                                          SCREEN: INSTALL
 These parameters define information which is used when INSTALLing Q-LINK.
 Changes to this information require a re-install of Q-LINK before they can
 take effect. Note: the CDB BDI must indicate an unused Alternate File
 Common Bank (AFCB) or Configured Common Bank (CCB) BDI. Enter a '?' in
 any field for detailed information.

 Q-LINK COMMON DATA BANK BDI                  0400707
 Q-LINK FIXED GATE SUB SYSTEM BDI
 Q-LINK COMMON DBANK NAME                     QCOMCB$D
 Q-LINK C-D-B START ADDRESS                   0700000
 CREATE SYS$LIB$*RUN$ RUNSTREAMS              TRUE
 MAPPER MASM PROCEDURE FILE (MRO)             NONE
 SUPUR FILE CODE FOR TXN SERVERS              19
 QLINK VALTAB NAME KEYWORD VALUE              QLNKKT

 INSTALL NAME FOR QLINK PROGRAM
 INSTALL NAME FOR QINDEX PROG
 INSTALL NAME FOR QMON PROGRAM
 INSTALL NAME FOR QLSIM PROGRAM
 INSTALL NAME FOR QLKDDP PROGRAM
```

The following screen is Selection 38 from the main menu. The parameter values shown were used to generate Q-LINK absolutes on the release tape.

```
                                                          SCREEN: BUILD
 These parameters define information which can only be changed by a Q-LINK
 system generation. Enter a question mark (?) in any field and transmit for
 information on how to configure that parameter. DMS1100 related parameters
 can be viewed with a selection from the main menu (use the BACK command).


 MAX VARIABLE NAMES                           400
 DATA STORAGE AREA (DOUBLE WORDS)             1000
 PROGRAM STORAGE SIZE (COMMANDS)              1800
 MAXIMUM LABELS IN A PROGRAM                  400
 MAXIMUM FILES (DEF F DIRECTIVES)             10

 Q-LINK DBANK START ADDRESS                   0154000
 DB4 PRODUCT LIBRARY FILE                     NONE
 DTM PROC FILE                                MAPTEST*HLDMAP
 DTM SCHDLR INTERFACE BDI                     0403133
 QWIZZ PRODUCT FILE
 DDP-PPC RELOCATABLE FILE
 DDP-PPC PROC FILE
```

Chapter 15: Installing Q-LINK across 2200 Systems

For sites wishing to generate Q-LINK on an 2200 system, and then install the generated system on another 2200, the following procedure may be used:

On the Generation System (S-1):

1. Follow the step outlined in this Q-LINK Installation Guide to REGISTER the Q-LINK release tape with COMUS (REGISTER is only required one time per release level).
2. Use the COMUS CONFIGURE to enter the Q-LINK Product License Information, for the system where the generation is to take place (S-1). If Q-LINK is going to execute on S-1, the information on the enclosed "Q-LINK Product License Key Information" sheet should be for S-1 and must be entered exactly as shown.

   Note: If the 2200/2200 system where the generation is taking place (S-1) is not licensed by KMSYS Worldwide for Q-LINK execution, you must enter the product license information as shown on the Product License Key Sheet with the exception of the following three parameters:

   - The "2200/2200 SYSTEM TYPE" must be the system type of the S-1 system instead of the one shown on the product key sheet
   - The "SITE ID CONFIGURED IN THE EXEC" must also be for S-1
   - The "Q-LINK PRODUCT VALIDATION KEY" must be set to "BUILD ONLY".
3. BUILD and INSTALL Q-LINK as shown in the Installation Guide.

   Note: The Q-LINK BUILD/INSTALL parameters (screens "ENVIRON", "INSTALL", and "BUILD") should be configured for the system on which Q-LINK will run (S-2). If Q-LINK is going to be run on S-1 and the configuration values for the BUILD and INSTALL are not the same for both machines, a separate BUILD and INSTALL must be done for Q-LINK execution on S-1.
4. If you do not transport your product libraries via the COMUS LIBSAVE runstream, skip to step 8 below.
5. Once the Q-LINK installation runstream has been successfully completed, update the Q-LINK Product License Information for S-2. You should have received a "Q-LINK Product License Key Information" sheet for S-2. The information must be entered exactly as shown.
6. Use the COMUS CONFIGURE to PROCESS the dynamic configuration parameters for S-2 as shown in the Q-LINK installation guide.
7. Transport product libraries via the COMUS LIBSAVE/LIBLOAD routines, and skip to step 17 below.

8. After doing a COMUS CONFIGURE, use the BOUT command to create an SDF data file containing the configuration set information from the COMUS database.

   Example:

   ```
   CONFIGURE
   BOUT SET=QLINK6R1 PRODUCT=QLINK LEVEL=6R1 FILE=QLINK*CONF-SET
   ```

9. Transport this file to the S-2 system by what ever means you have available (DDP, tape, etc.).

On the Execution System (S-2):

10. Use the COMUS REGISTER to register the output tape from the Q-LINK generation on S-1 (REGISTER only required one time per release level).
11. Use the COMUS CONFIGURE command and enter "\NOSCR" (no screen mode).
12. @ADD the SDF data file transported from S-1.
13. Enter "\SCR" to return to full screen mode.
14. Update the Q-LINK Product License Information for S-2. You should have received a "Q-LINK Product License Key Information" sheet for S-2. The information must be entered exactly as shown.
15. Do a COMUS INSTALL using the output from the Q-LINK generation on S-1. Follow the steps for Q-LINK installation as shown in the installation guide.
16. Once the Q-LINK installation runstream has been successfully completed, use the COMUS CONFIGURE to PROCESS the dynamic configuration parameters for S-2 as shown in the Q-LINK installation guide.
17. STEP-6 through STEP-9 (installing into BIS, QUTIL registration, QLINK$DTM registration and VALTAB creation) must be done for S-2 as outlined in the Q-LINK installation guide.

Chapter 16: Applying Changes to Q-LINK

If any changes are to be applied to either the generation skeletons or any of the system components (i.e., you have received a fix, not included in the release, from KMSYS Worldwide), this chspter should be followed before starting the COMUS BUILD; otherwise, you will not need this chapter.

16.1 COMUS Change Format

The following procedure can be used if the fix you receive is in COMUS change format, ready to be applied to Q-LINK using the COMUS batch insert before the BUILD (this is accomplished by simply adding the change element):

```
▶@qual qkms
▶I:002333 QUAL complete.
▶@comus qlink
▶COMUS 6R8D (060816 1323:09) 2007 Apr 18 Wed 1321:58
▶Copyright (c) 1995-2006 Unisys Corporation.
▶COMMAND ?   ▶@add pf.qlink/chg
▶************** CHANGE KMS00001 INSERTED IN DATABASE
▶UPDATING ACCESS FILES ...
▶ACCESS FILES HAVE BEEN UPDATED
▶INSERT TASK COMPLETED ************************
▶demand
▶INSERT QLINK CHANGE (SITEID = KMS) SCREEN MODE ? ▶exit
▶END COMUS
```

16.2 TCF Change Format

If the fix you receive is in TCF format and not COMUS change format, you can use the following COMUS full-screen INSERT to apply the change:

```
▶@qual qkms
▶I:002333 QUAL complete.
▶@comus qlink
▶COMUS 6R8D (060816 1323:09) 2007 Apr 19 Thu 1233:57
▶Copyright (c) 1995-2006 Unisys Corporation.
▶COMMAND ?   ▶insert
▶INSERT QLINK CHANGE (SITEID = KMS) SCREEN MODE ? ▶
```

Simply transmitting on the previous query will place you in full-screen mode. All the required information is filled in as is shown in the following example:

```
** CHANGE **     PRODUCT :  QLINK          CHG-NUMBER :
PROGRAMMER   :  Q-LINK Installer          CLASS1     :  fix
APPLY-START  :                            STATUS     :
APPLY-END    :                            APPLIES-TO :
MERGE-SEQ    :
REL-CHANGES  :
UCFS         :
FEATURE      :
KEYWORDS     :  example

DESCRIPTION  :  This is an example of a COMUS full-screen INSERT with
                the change having been placed into the TCF formatted
                element:

                           qlinkpf.qlink/tcf

                                                    COMMAND (I,R,C,D) :
                    FIELDS MARKED WITH ◤ MUST BE FILLED IN
```

Once the change has been properly documented on the previous screen, the actual code can be applied. Here it is shown coming from an element in a program file:

```
▶CODE           : ▶@add pf.Q-LINK/tcf
▶CODE           : ▶
▶************** CHANGE KMS00013 INSERTED IN DATABASE
▶UPDATING ACCESS FILES ...
▶ACCESS FILES HAVE BEEN UPDATED
▶INSERT TASK COMPLETED ************************
▶INSERT QLINK CHANGE (SITEID = KMS) SCREEN MODE ?  ▶list
▶LIST QLINK CHANGE ?   ▶select num=kms13
▶#1 NUM : KMS00013                          SELECT LIST:      1
▶LIST DIRECTIVE ? ▶cview 1
```

Also, notice that it was decided to display the change in the COMUS database by using the COMUS commands: LIST, SELECT and CVIEW. The following is a sample of how the output screen might appear:

```
CHG-NUMBER      : KMS00013      STATUS      : TEST
KEYWORDS        : EXAMPLE
DESCRIPTION     :
          This is an example of a COMUS full-screen INSERT with
          the change having been placed into the TCF formatted
          element:

                     qlinkpf.qlink/tcf
ELEMENTS        : ELEMENT
PRODUCT         : QLINK         ORG         :              DATE        : 18 JUN 07
CLASS1          : FIX           UPDATE-NUM  : 0            UPDATE-DATE: 18 JUN 07
CLASS2          : MINOR         INT-LEVEL   :              INT-DATE    :
CLASS3          : MAJOR         APPLY-START:               APPLY-END   :
STATUS          : TEST          MERGE-SEQ   : 000
SORT-SYSNUM     :
FEATURE         :
PROGRAMMER      : Q-LINK INSTALLER
LINES OF CODE   : 3
CODE            :
:*ELEMENT
:-1000,1000   element
:          first code line at 1000
▸VIEW DIRECTIVE ?  ▸
```

Index

If you would like to help us make our documentation better, please take a few moments to complete this form and return it to KMSYS Worldwide. We are always looking for ways to improve our products and your feedback will help us reach our goal.

Name _____

Company _____

Address _____

_____

City _____ State/Province _____

Country _____ Zip/Mail Code _____

Document Name _____ OS Level _____

KMSYS Worldwide Product _____ Level _____

Please rate the documentation on a scale of 1 to 5:

	5	4	3	2	1
Complete	❑	❑	❑	❑	❑ Incomplete
Accurate	❑	❑	❑	❑	❑ Inaccurate
Usable	❑	❑	❑	❑	❑ Unusable
Readable	❑	❑	❑	❑	❑ Unreadable
Understandable	❑	❑	❑	❑	❑ Unintelligible
Attractive	❑	❑	❑	❑	❑ Unattractive
Excellent	❑	❑	❑	❑	❑ Poor

What information did you expect to find that was omitted?

_____

Is more information needed? ❑ Yes ❑ No. If yes, on what topic?

_____

Did you find factual errors in the documentation? ❑ Yes ❑ No. If yes, please give page number and description of the error.

_____

If the documentation is difficult to understand, please specify page number and problem.

_____

Is the documentation intimidating? ❑ Yes ❑ No.

_____

Are the manuals: ❑ Too long? ❑ Too short? ❑ About the right length?

_____

Other suggestions or comments? (Use back of form if necessary.)

_____

(Additional Comments)

. . . . . Fold along dotted line. . . . . .

KMSYS
WORLDWIDE, INC
P.O. Box 669695
Marietta, GA 30066
U.S.A.

Attn: Technical Documentation Section