5

Configuring and Running a Multiclient Remote Environment Server (MRES) with TCP/IP

This chapter tells how to configure a Multiclient Remote Environment Server with TCP/IP (MRES with TCP/IP) on a network that has the following software:

- OS/390 Version 2.4, or a later version
- IBM Transmission Control Protocol (TCP/IP) for MVS Version 3 Release 2 (5655–HAL) with PTF UN98840, or equivalent.
- Service Desk for OS/390 Version 1.2

You may need other software, depending on the configuration of your network.

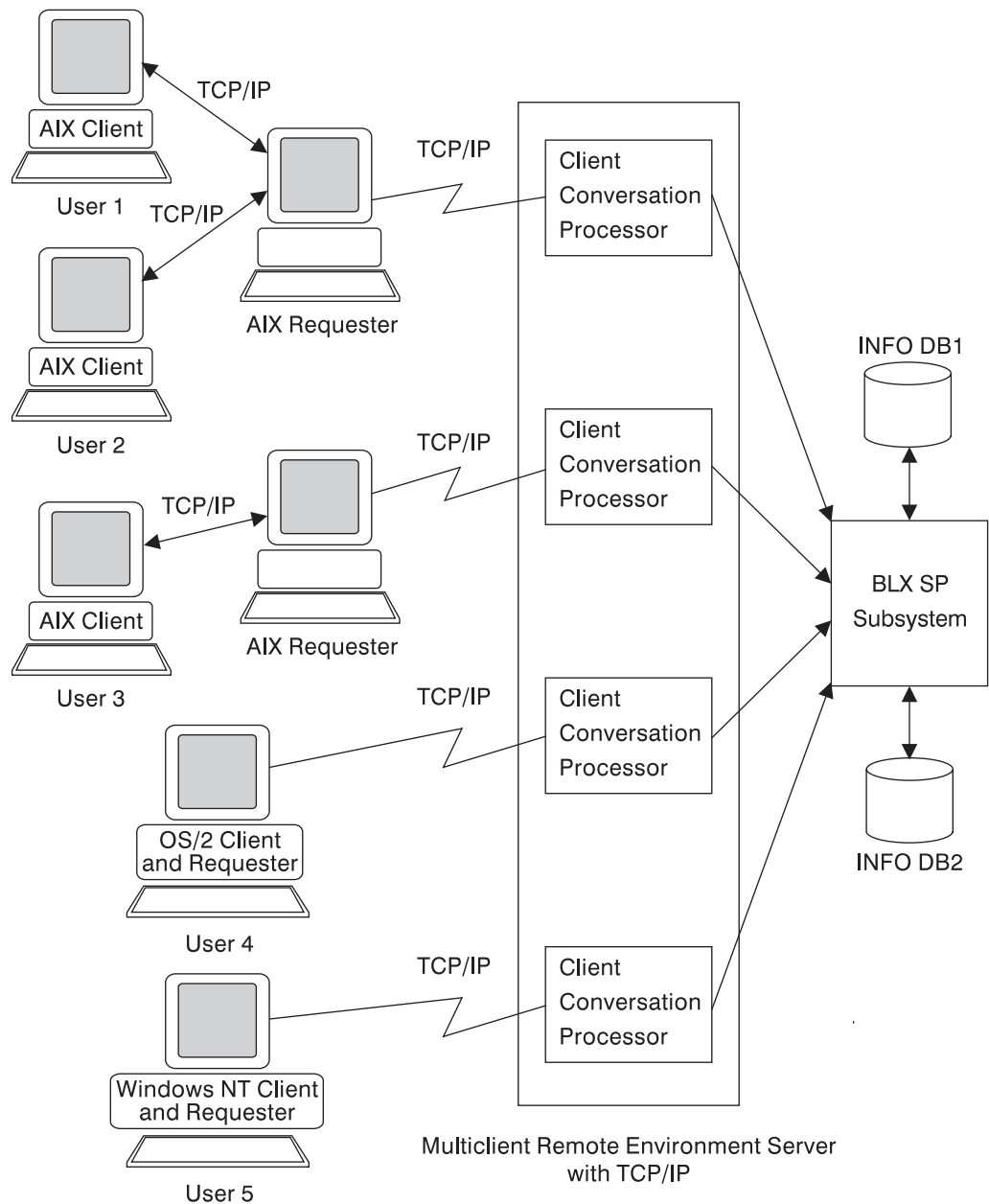Figure 23 on page 5-2 illustrates an MRES with TCP/IP.

Figure 23. MRES with TCP/IP and Multiple Client Conversation Processors Started

Using an MRES with TCP/IP

The systems programmer creates one or more MRES with TCP/IP cataloged procedures in **SYS1.PROCLIB**. Each procedure contains the MVS JCL required to start the MRES with TCP/IP address space. The systems programmer must also create a data set pointed to by the procedure which contains the values for MRES with TCP/IP startup parameters.

The systems programmer works with the security administrator and the TCP/IP communications programmer to determine the appropriate security mechanisms to use for securing the MRES with TCP/IP and the transaction programs it serves.

The Service Desk for OS/390 systems programmer then works with the TCP/IP communications programmer to define resources for the MRES with TCP/IP.

While the TCP/IP definitions are being done, the security administrator defines the agreed upon security classes and profiles. For example, when using RACF as the security product with OpenEdition, you need to add the MRES started procedure name to the **RACF STARTED** class, specifying the following information for **STDATA**:

> **USER**= specify a userid that has an OMVS segment.

> **Note:** You may consider using an OMVS default user. In the *Open Edition Planning Guide*, see the section entitled "Setting Up Default OMVS Segments" for additional information on creating an OMVS default user.

> **TRUSTED***(NO)*

> **PRIVILEGED***(YES)*

The commands to perform this are:

```
SETROPTS GENERIC(STARTED) RDEFINE STARTED MR*.* STDATA(USER(OEDFLT) TRUSTED(NO)
              PRIVILEGED(YES)

SETROPTS RACLST(STARTED) REFRESH
```

In a RACF environment, the first **SETROPTS** command causes the OEDFLT user ID to be assigned to a started task which has a name beginning with *MR*. The second **SETROPTS** command performs a refresh of the RACF list.

After the cataloged procedure, MRES parameters, TCP/IP, and security definitions are complete, the system programmer and the communications programmer verify that each client user ID can communicate with the MRES with TCP/IP and exchange data.

The operator starts an MRES with TCP/IP address space with the MVS **START** command, described in "START Command Syntax" on page 5-14. The MRES parameter **PORT** specifies the unique port number for TCP/IP to use to communicate with the MRES address space. The MRES parameter **IPADDRESS** specifies the IP address used to access the MRES address space. The MRES parameter **MAXCONNECT** specifies the maximum number of client conversations (connections) that the MRES can process concurrently. Additional address spaces for MRES for TCP/IP communication can also be started. Each address space must have a unique IP address and port number combination.

Note: You can start multiple MRESs from the same procedure. Remember that the **PORT** number for each MRES started must be unique. To start multiple MRESs, specify the name of the cataloged procedure in the **START** command, and ensure that the **PRM**= parameter contains different parameter data; the **PORT** number must be different for each MRES. The name of an MRES address space is the same name as that of the catalogued procedure that started it unless the **JOBNAME**= parameter is specified.

When the MRES address space receives a communication request, it grants the request provided it has not reached its maximum number of communication connections. When a client ends its last or only HLAPI session, the communication connection is ended. If the MRES receives a request and all its communication connections are being used, the request is rejected.

The operator uses the MVS **STOP** command to stop an MRES address space. Upon receipt of the stop command, a write-to-operator (WTO) message is sent saying how many client connections are still active. After the **STOP** command is issued, the MRES does not accept any more connection requests. After a period of time (specified in a startup parameter) expires or when the MRES has no more client connections, the address space ends. Additional WTO messages may be sent at a specified time interval while the MRES is waiting to end.

MRES Configuration Tasks

To configure an MRES with TCP/IP, do the following tasks:

1. Plan your configuration. "Planning Your MRES with TCP/IP Configuration" discusses this task.

2. Define a cataloged procedure for the MRES with TCP/IP. "Defining a Procedure for an MRES with TCP/IP" on page 5-5 discusses this task.

3. Create a data set and specify the MRES parameters. This data set is referenced by the **BLMYPRM DD** statement in the catloged procedure; "Coding the Parameters for MRES with TCP/IP" on page 5-6 describes this task.

4. Add the data sets that contain the MRES load modules to the authorized program facility (APF) library. "Adding the Data Sets to the APF List" on page 5-14 discusses this task. If the data sets are specified in the link list, or if you have them in the link pack area, you can skip this step.

5. Implement your security plan. Refer to your security product's documentation for information on this task.

When you are ready to start an MRES with TCP/IP, see "Starting and Stopping an MRES with TCP/IP" on page 5-14 for the syntax.

Planning Your MRES with TCP/IP Configuration

Setting up an MRES is independent of setting up a client. You need to consider which clients will be using the MRES and whether you want them all to use the same MRES or whether you want each client to use a different MRES. You will need to tell the clients what TCP/IP port numbers and addresses to use. See "Choosing a Server" on page 2-1 for a discussion of other factors you should consider while planning your configuration.

Defining a Procedure for an MRES with TCP/IP

You must define a procedure for an MRES with TCP/IP in **SYS1.PROCLIB**.
Figure 24 lists the JCL statements to include in an MRES with TCP/IP procedure.
This JCL is shipped as **BLMMRES** in the **SBLMSAMP** sample library.

```
//**************************************************************************
//*
//*  THIS SAMPLE CATALOGUED PROCEDURE STARTS THE MULTICLIENT REMOTE
//*  ENVIRONMENT SERVER (MRES) ADDRESS SPACE USING THE SAMPLE
//*  PARAMETERS MEMBER.
//*
//*  CHANGE THE NAME CODED ON THE DSN KEYWORD OF THE BLMYPRM DD
//*  STATEMENT TO SPECIFY YOUR OWN MRES PARAMETERS DATA SET.
//*  IF YOU ARE USING A PARTITIONED DATA SET FOR YOUR PARAMETERS,
//*  CHANGE THE PRM= ASSIGNMENT VALUE ON THE PROC STATEMENT TO YOUR
//*  DEFAULT MEMBER NAME.
//*
//*  STORE THE UPDATED PROCEDURE AS A MEMBER IN YOUR SYSTEM'S PROCEDURE
//*  LIBRARY, SUCH AS SYS1.PROCLIB.
//*
//**************************************************************************
//BLMMRES   PROC PRM=BLMMRESP                    PARAMETER DATA MEMBER
//MRESPGM   EXEC PGM=BLMYSM00,REGION=6M,TIME=1440
//STEPLIB   DD   DISP=SHR,DSN=BLM.V1R2M0.SBLMMOD1        APF AUTHORIZED
//BLMYPRM   DD   DISP=SHR,DSN=BLM.V1R2M0.SBLMSAMP(BLMMRESP)    PARMS
//APIPRINT  DD   SYSOUT=*                                 LLAPI LOG
//HLAPILOG  DD   SYSOUT=*                                 HLAPI LOG
//         PEND
```

Figure 24. Example MRES withTCP/IP Cataloged Procedure

A description of the JCL statements follows:

BLMMRES

This is the name of the instream catalogued procedure.The **PRM=** parameter
specifies the name of a data set member that contains the parameters unique to
this MRES. It is set to a default value of **BLMMRESP**, the name of the
MRES parameter data set supplied in the samples data set. If **PRM=** is not
specified, the values contained in the sample parameter data member
BLMMRESP are used. To override the value contained in the cataloged
procedure, you can specify a **PRM=** on the **START** command.

MRESPGM

This statement specifies the program to be processed, its region size, and how
long it can run. The value for **PGM=** must be *BLMYSM00* to call the MRES
program. The **REGION=** parameter specifies how much memory to reserve
for the MRES program. The larger the region size, the more HLAPI sessions
can run concurrently. Declaring **TIME=***1440* tells the operating system not to
time out the MRES. If this parameter and value are not specified, the MRES
will receive an out-of-time ABEND.

STEPLIB

This statement specifies the program libraries used by the MRES. You must specify the data sets in a **STEPLIB DD** statement if you do not specify them in the link list or if you do not have the load modules in the link pack area. The data sets must be authorized program facility (APF) libraries.

BLMYPRM

This statement specifies the data set or PDS member containing the MRES parameters. Both fixed and variable length record formats are supported.

APIPRINT

This statement specifies the LLAPI log. If any messages about LLAPI transaction activity are generated, they are to be sent to the job's **SYSOUT** class. Refer to the *Tivoli Service Desk for OS/390 Application Program Interface Guide* for more information about these messages.

HLAPILOG

This statement specifies the HLAPI log. If any messages about HLAPI transaction activity are generated, they are to be sent to the job's **SYSOUT** class. The *Tivoli Service Desk for OS/390 Application Program Interface Guide* contains additional information about these messages.

Coding the Parameters for MRES with TCP/IP

You must specify the MRES parameters in the data set or PDS member that you specify on the **BLMYPRM DD** statement. A sample that you can use for your MRES parameters is shipped as **BLMMRESP** in the **SBLMSAMP** library; this is described in "Sample MRES Parameters" on page 5-13.

Several basic rules must be followed when you code the parameter data:

- Comments must begin with /* and end with */

- Comments can be in any column between 1 and 72, inclusive

- Nothing (except comments) can be in column 1

- Nothing can be in any column greater than column 72

- Begin the parameters data with a statement identifier of *BLMYPRM*

- You can separate parameters with a comma. Parameters need not appear on separate lines

- The final parameter should end with a semicolon

These are the MRES parameters that you can specify in your **BLMYPRM** data set:

```
[APMINSTRUMENT={OFF|ON}]
[MAXCONNECT={10|n}]
SHUTDOWNTRY=hhmmss
SHUTDOWNWT=hhmmss
[WRITEOPER={1|code}]
COMMTYPE=TCPIP
[IPADDRESS={0.0.0.0|n.n.n.n}]
[PORT={1451|n}]
[PRESTARTSESSIONS={NO|YES}]
[APPLICATION_ID=id]
[PRIVILEGE_CLASS=id]
[SESSION_MEMBER=id]
[APIMSG_OPTION={C|P|B}]
[BYPASS_PANEL_PROCESSING={NO|YES}]
[CLASS_COUNT={0|n}]
[DATABASE_ID={5|n}]
[DEFAULT_DATA_STORAGE_SIZE={1024|n}]
[DATE_FORMAT={DATABASE|pattern}]
[DEFAULT_OPTION={ALL|REQUIRED|NONE}]
[HLIMSG_OPTION={C|P|B}]
[PDB_TRACE={NO|YES}]
[SPOOL_INTERVAL={0|n}]
[TABLE_COUNT={0|n}]
[TIMEOUT_INTERVAL={300|n}]
```

Figure 25. BLMYPRM MRES Parameters

The values that you can specify for these parameters are as follows:

APMINSTRUMENT={OFF|ON}

This parameter specifies the APM Instrumentation flag. A specification of **ON** indicates that the MRES should generate heartbeats or pulses to allow Tivoli Global Enterprise Manager (GEM) Application Policy Manager (APM) to monitor the status of the MRES. To have pulses generated, specify **APM=ON**. If a value is not specified, the default value is **OFF**. This parameter is optional. This parameter is provided for those users who are using Tivoli Global Enterprise Manager (GEM) to manage Service Desk for OS/390 in a Tivoli Management Environment. The *Tivoli Service Desk for OS/390 Guide to Integrating with Tivoli Applications* contains additional information about the Global Enterprise Manager.

If you are not using Service Desk for OS/390 in a Tivoli GEM environment, you can omit this parameter.

MAXCONNECT={10|n}

This parameter specifies the maximum number of client conversation processors this MRES can run concurrently. Valid values are from 1 to 50. The default value is 10 This is an optional parameter.

SHUTDOWNTFY=hhmmss

A required parameter that specifies the time interval between shutdown notification messages. The value specified indicates the amount of time between operator notification messages that follow the first message. When the operator issues the MVS **STOP** command, the MRES sends a message to the operator indicating the number of conversations that are still connected. After

this first notification message is sent, notification messages are sent at the interval specified on this parameter. For example, if you specify a value of 000000, you will receive only one notification message. If you specify a value of 000500, you will receive the first notification message and additional messages at 5-minute intervals until either the **SHUTDOWNTFY** time expires or all the conversations have ended.

You must specify this interval as **HHMMSS** where

HH	Hours
MM	Minutes
SS	Seconds

SHUTDOWNWT=hhmmss

A required parameter that specifies the shutdown wait time period. The value specified indicates a period of time that the MRES is to continue processing after receiving the MVS **STOP** command. This time period permits client conversations to complete any processing that was active when the operator issued the **STOP** command. Additional conversations are not accepted during this time, but those already connected can continue until the time expires. If no clients are connected when the **STOP** command is issued, the MRES stops processing immediately regardless of the interval specified on this parameter. Also, when all the conversations have stopped, the MRES stops processing immediately.

You must specify this interval as **HHMMSS** where

HH	Hours
MM	Minutes
SS	Seconds

WRITEOPER={<u>1</u>|code}

This parameter specifies the default write-to-operator (WTO) routing code. Valid values are from 1 to 128. If not specified, the default value is 1. All WTOs that are not a result of command responses are automatically routed to this code. To determine the routing codes for a console, you can do one of the following:

- Display console characteristics by issuing the *DISPLAY CONSOLES,A* command from the console.

- Review the **ROUTCODE** parameter of the **CONSOLE** statements in the **CONSOL.***xx* member of **SYS1.PARMLIB**.

This parameter is optional.

COMMTYPE=TCPIP

A required parameter that specifies the communications protocol to be used. You must specify **TCPIP** for the MRES with TCP/IP.

IPADDRESS={<u>0.0.0.0</u>|n.n.n.n}

This parameter specifies the Internet or IP address of the network interface for accessing the MRES with TCP/IP. It is the unique address of the host on a

network and is specified in dotted decimal format. This consists of four numbers with valid values from 0 to 255, separated by periods. Some hosts have more than one network address. If you want to allow this MRES to receive connection requests from any of the network interfaces, specify the dotted decimal string of all zeros (000.000.000.000). If a value is not specified, the default value is *0.0.0.0* (this is the equivalent of 000.000.000.000).

PORT={1451|n}

If you specified **COMMTYPE=TCPIP**, this parameter specifies the unique port number used to communicate between TCP/IP and the MRES address space. This number must not be used by any other application, including another MRES. Valid values are from 1 to 65534. The value you specify here becomes the default port number for this MRES address space. This parameter is optional. If a value is not specified, the default value is *1451*.

PRESTARTSESSIONS={NO|YES}

Pre-start API sessions indicator. Set this to **YES** to specify that the API sessions are to be pre-started. If you specify **NO** or omit this parameter, the API sessions are not pre-started. This parameter is optional.

APPLICATION_ID=id

Contains a 1- to 8-character uppercase application ID that Service Desk for OS/390 uses for this session. The application ID is specified on the HL01 transaction and can be specified on many other HLAPI transactions, so it can vary over the life of the HLAPI session. The ID must be an eligible user of the privilege class being used. This keyword is required when **PRESTARTSESSIONS=YES** is specified; otherwise, it is optional.

PRIVILEGE_CLASS=id

Contains a 1- to 8- byte privilege class name, which can contain DBCS characters enclosed by an SO/SI pair. A privilege class remains in effect until your application specifies a different privilege class name. An application can specify an initial privilege class that grants all authority required for the duration of the Service Desk for OS/390 session. This keyword is required when **PRESTARTSESSIONS=YES** is specified; otherwise, it is optional.

SESSION_MEMBER=id

Contains a 7- or 8-character load library session parameter member name that Service Desk for OS/390 uses for this session. Session member names begin with the character string *BLGSES* and cannot contain imbedded blanks. This keyword is required when **PRESTARTSESSIONS=YES** is specified; otherwise, it is optional.

APIMSG_OPTION={C|P|B}

Contains a 1-character LLAPI message option parameter P, C, or B.

- A value of **P** specifies that the LLAPI writes messages to the **APIPRINT** data set.

- A value of **C** specifies that the LLAPI chains messages and passes them from the LLAPI to the HLAPI for conversion into message PDBs.

| ■ A value of **B** specifies that the LLAPI performs both **P** and **C**.

If you omit this parameter, then the LLAPI performs option **C**. This parameter is used only if **SPOOL_INTERVAL** is specified and is not set to zero. This parameter is optional and is used only if **PRESTARTSESSIONS=YES**.

BYPASS_PANEL_PROCESSING={<u>NO</u>|YES}
Bypass panel processing indicator. Set this to **YES** to specify that no panels be used in record processing other than those used by the delete transaction. If you specify **NO** or omit this parameter, the HLAPI performs panel processing.

If you specify **BYPASS_PANEL_PROCESSING = YES**, you must use data model records for the following transactions:

■ HL08 Create record

■ HL09 Update record

■ HL12 Add record relation

This parameter is optional and is used only if **PRESTARTSESSIONS=YES**.

CLASS_COUNT=<u>0</u>|n
Contains a character number that indicates the maximum number of Service Desk for OS/390 privilege class records that can be maintained in storage during the life of this Service Desk for OS/390 session. If you omit this parameter or enter zero as its value, the Service Desk for OS/390 session operates with a single privilege class record in storage at a time. This parameter is optional and is used only if **PRESTARTSESSIONS=YES**.

DATABASE_ID=<u>5</u>|n
A 1-byte character field containing the 1-character ID number of the database to be used. For Service Desk for OS/390 records, the database ID number is 5. If you omit this parameter, the HLAPI automatically sets the database ID to 5. This parameter is optional and is used only if **PRESTARTSESSIONS=YES**.

DEFAULT_DATA_STORAGE_SIZE=<u>1024</u>|n
Contains a character specifying how much additional storage is allocated to hold default response data from an alias table when your application is creating records. When the HLAPI creates records, it calculates the size of the response buffer it needs by totaling the lengths of all the input data PDBs and adding the specified default data storage size. If you omit the default data storage size, the HLAPI adds a default of 1024 bytes. When the HLAPI performs create response processing, it always checks to make sure the response will not overlay storage. If the response will overlay storage, the HLAPI transaction will end with an error code. You use this parameter with the **DEFAULT_OPTION** parameter. This parameter is optional and is used only if **PRESTARTSESSIONS=YES**.

DATE_FORMAT=<u>DATABASE</u>|pattern
Contains a character field that specifies how your application uses dates. If the value **DATABASE** is specified, the default format from the database is

assumed. If you omit this parameter, the value **DATABASE** is assumed. Valid
values are:

    MM/DD/YY
    MM-DD-YY
    MM.DD.YY
    MM/DD/YYYY
    MM-DD-YYYY
    MM.DD.YYYY
    DD/MM/YY
    DD-MM-YY
    DD.MM.YY
    DD/MM/YYYY
    DD-MM-YYYY
    DD.MM.YYYY
    YY/MM/DD
    YY-MM-DD
    YY.MM.DD
    YYYY/MM/DD
    YYYY-MM-DD
    YYYY.MM.DD
    DDMMMYY
    DDMMMYYYY
    YYDDD
    YYYYDDD

This parameter is optional and is used only if **PRESTARTSESSIONS=YES**.

DEFAULT_OPTION=ALL|REQUIRED|NONE
Contains a character field that specifies how the HLAPI performs create
default data response processing in this session. The valid data values for
DEFAULT_OPTION are **ALL**, **REQUIRED**, and **NONE**. **ALL** specifies
that all response fields specified in a PIDT are candidates for default
responses. **REQUIRED** specifies that only required fields are candidates for
default responses. The HLAPI does not perform default response processing if
you omit this field or specify it as **NONE**. After the create transaction
completes, the HLAPI reverts to the initial default specification for record
creation unless overridden again. This parameter is optional. If you omit this
parameter, the value **NONE** is assumed. This parameter is optional and is used
only if **PRESTARTSESSIONS=YES**.

HLIMSG_OPTION=C|P|B
Contains a 1-character HLAPI message option parameter **P**, **C**, or **B**.

- A value of **P** specifies that the HLAPI writes messages to the
  HLAPILOG data set.

- A value of **C** specifies that the HLAPI chains messages on the PDB
  message chain.

- A value of **B** specifies that the HLAPI performs both P and C.

If you omit this parameter, then the HLAPI performs option **C**. The HLAPI writes messages passed back from the LLAPI to the **HLAPILOG** data set. This parameter is used only if **SPOOL_INTERVAL** is specified and is not set to zero. This parameter is optional and is used only if **PRESTARTSESSIONS=YES**.

PDB_TRACE=NO|YES

This parameter specifies whether the HLAPI should perform PDB data tracing for debugging purposes or PDB data logging to the HLAPILOG output file. Setting this parameter value to **YES** causes the logging of up to 32 bytes of PDBDATA information for each PDB used throughout the session. A value of **NO** causes no PDB logging to be performed. This parameter is optional and is used only if **PRESTARTSESSIONS=YES**. If you omit this parameter, the value **NO** is assumed.

SPOOL_INTERVAL=0|n

Contains a character specifying the number of minutes that the HLAPI spools the activity logs **HLAPILOG** and **APIPRINT** when messages are printed. If the HLAPI is spooling to a data set and this time interval has passed, the activity logs are recycled and new log information is written starting at the top of the data set, writing over any existing information. If you omit this parameter, the HLAPI does not log messages and the settings in **APIMSG_OPTION** and **HLIMSG_OPTION** are ignored. This parameter is optional and is used only if **PRESTARTSESSIONS=YES**.

TABLE_COUNT=0|n

Contains a character that indicates the maximum number of alias tables and PIDTs and anchored PIPTs that the HLAPI can maintain in storage during the life of a Service Desk for OS/390 session. Static PIDTs and PIDTs generated from data view records are treated the same for caching purposes. It can take a significant amount of time to generate a PIDT from data view records. The length of time depends on the number of data attribute records (and validation records they reference) contained in the data view record. Therefore, it can be especially important to direct the HLAPI to maintain PIDTs in storage if you are using data models. If you specify this value as zero or omit it, the Service Desk for OS/390 session will not process **ALIAS_TABLE** parameters or cache PIDTs. Alias table and PIDT processing can increase transaction run time due to the increased I/O time of loading and unloading tables. By balancing the table count to alias table and PIDT usage, you can reduce to zero the additional I/O overhead for long-running applications. This parameter is optional and is used only if **PRESTARTSESSIONS=YES**.

TIMEOUT_INTERVAL=300|n

Contains a character specifying the number of seconds that a transaction can run before a timer interrupt occurs. If you specify a value between 0 and 45 seconds, the HLAPI uses a value of 45 seconds. If you specify a value of 0 or omit this parameter, the HLAPI uses a default value of 300 seconds (five minutes). This parameter is optional and is used only if **PRESTARTSESSIONS=YES**.

Sample MRES Parameters

A sample of the MRES parameters is provided in **BLMMRESP** in the
SBLMSAMP data set. If you rename this member or create a new one, be sure to
change the member name in the **BLMYPRM DD** statement.

```
BLMYPRM                              /* SPECIFY MRES PARAMETERS                 */
/******************************************************************************/
/*                                                                            */
/*             PARAMETERS TO CONTROL THE GENERAL MRES SESSION                 */
/*                                                                            */
/******************************************************************************/

  APMINSTRUMENT=OFF,                 /* APM INSTRUMENTATION (ON OR OFF)   */
  MAXCONNECT=10,                     /* MAXIMUM NUMBER OF CONNECTIONS     */
  SHUTDOWNTFY=000200,                /* SHUTDOWN NOTIFY TIME (HHMMSS)     */
  SHUTDOWNWT=000500,                 /* SHUTDOWN WAIT TIME (HHMMSS)       */
  WRITEOPER=1,                       /* WRITE-TO-OPERATOR ROUTING CODE    */

/******************************************************************************/
/*                                                                            */
/*             PARAMETERS TO CONTROL THE COMMUNICATIONS SESSION               */
/*                                                                            */
/******************************************************************************/

  COMMTYPE=TCPIP,                    /* COMMUNICATIONS (APPC OR TCPIP)    */
  IPADDRESS=000.000.000.000,         /* INTERNET PROTOCOL ADDRESS         */
  PORT=1451,                         /* TCP/IP UNIQUE PORT NUMBER         */

/******************************************************************************/
/*                                                                            */
/*             PARAMETERS TO CONTROL THE PRE-STARTING OF API SESSIONS         */
/*                                                                            */
/******************************************************************************/

  PRESTARTSESSIONS=NO,               /* PRE-START (YES OR NO)             */
  APPLICATION_ID=XXXXXXXX,           /* APPLICATION ID                    */
  PRIVILEGE_CLASS=XXXXXXXX,          /* PRIVILEGE CLASS NAME              */
  SESSION_MEMBER=BLGSESXX,           /* SESSION MEMBER NAME               */
  APIMSG_OPTION=C,                   /* LLAPI MESSAGES (C, P, OR B)       */
  BYPASS_PANEL_PROCESSING=NO,        /* BYPASS PANELS (YES OR NO)         */
  CLASS_COUNT=0                      /* MAXIMUM NUMBER TO BE CACHED       */
  DATABASE_ID=5,                     /* DATABASE ID                       */
  DEFAULT_DATA_STORAGE_SIZE=1024,    /* DEFAULT SIZE IN BYTES             */
  DATA_FORMAT=DATABASE,              /* FORMAT OF DATE DATA               */
  DEFAULT_OPTION=NONE,               /* (ALL, REQUIRED, OR NONE)          */
  HLIMSG_OPTION=C,                   /* HLAPI MESSAGES (C, P, OR B)       */
  SPOOL_INTERVAL=0,                  /* MAXIMUM TIME IN MINUTES           */
  TABLE_COUNT=0,                     /* MAXIMUM TABLE ENTRIES             */
  TIMEOUT_INTERVAL=300;              /* MAXIMUM TIME IN SECONDS           */
```

Figure 26. Sample MRES Parameters

Adding the Data Sets to the APF List

If you neither specified the data sets that contain the MRES load modules in the link list nor have them in the link pack area, you must add the data sets to the APF library.

To define the data sets that contain the load modules as APF libraries, make an entry for each data set in the appropriate **PROG***xx* members of **SYS1.PARMLIB**. Each entry in a **PROG***xx* member includes the data set name (*dsn*) and the volume serial number (*volser*) of the library.

The changes will be activated the next time you IPL the system, or you can use the **SETPROG=***xx* command to dynamically activate the changes.

Starting and Stopping an MRES with TCP/IP

TCP/IP/MVS and the BLX-SP server must be running before you start an MRES with TCP/IP.

To start an MRES with TCP/IP, issue the MVS system operator **START** command for the cataloged procedure. To stop an MRES with TCP/IP, issue the MVS system operator **STOP** command. For full descriptions of the MVS **START** and **STOP** operator commands, refer to *OS/390 Operations: System Commands*. The **START** command accepts the parameters listed in the following section.

START Command Syntax

The syntax of the MVS system operator **START** command for an MRES with TCP/IP is as follows:

```
S blmmres  [,JOBNAME=job_name]
           [,prm=member_name]
```

S

The MVS system operator **START** command.

blmmres

The name of the cataloged procedure for this MRES with TCP/IP.

Note: The name of the cataloged procedure becomes the name of the job unless it is overridden by the **JOBNAME=** parameter.

JOBNAME=*job_name*

The value assigned to this keyword becomes the name of the newly started MRES. If this parameter is not specified, the name of the cataloged procedure becomes the name of the job.

Note: For the MRES with TCP/IP, multiple MRESs can be started with the same or different job names, but only with different parameter data. This is because the **PORT=** parameter must be unique for each started MRES.

prm=*member_name*

In the sample procedure **BLMMRES** (see Figure 24 on page 5-5), **prm=** specifies the name of a parameter data set member. This member should

| contain the parameters unique to this MRES. If **prm=** is not specified, the
| values contained in the sample parameter data member **BLMMRESP** are used.
| To override the value contained in the cataloged procedure, specify prm= on
| the **START** command.

STOP Command Syntax

The syntax of the MVS system operator **STOP** command for an MRES with
TCP/IP is as follows:

```
P job_name  [,A=n]
```

Note: Note that *job_name* is the name of the cataloged procedure that was
started, unless the parameter **JOBNAME** was specified on the **START**
command.

P

The MVS system operator **STOP** command.

job_name

The name of the cataloged procedure for this MRES with TCP/IP.

A=_n_

An optional parameter that specifies the address space number consisting of
from 1 to 4 hexadecimal digits (0–F). You can obtain this number in several
ways:

- Use the Display Jobs operator command (*D J,BLMMRES* displays
  information about all jobs with the name **BLMMRES**).

- Look for **ASID=**_n_ in message **BLM03170I**. This message is written to
  the console when the MRES is successfully started.

- Use other display commands. (See "Using MVS Operator Commands" on
  page 6-1 for information about the **DISPLAY** command.)

If you do not use this parameter, all jobs with the name **job_name** are
stopped. If you do use this parameter, only the job you specify is stopped.

Determining Values Clients Need

When you set up the communication links on the clients, you will need the
following values:

- Port number

  | This is the unique port number for a particular MRES with TCP/IP. For the
  | HLAPI/UNIX client system, this value must be set in the **/etc/services** file.
  | For the HLAPI/USS client system, this value must be set in either the
  | **/etc/services** file or the **hlq.ETC.SERVICES** data set. For HLAPI/2 and
  | HLAPI/NT client systems, this value must be set in the **SERVICES** file in
  | the **ETC** subdirectory.

- Internet address

This is the address of the MVS host with which you are communicating. If you use a host name in your database profile to identify the MVS host, this value must be set in the file on the HLAPI/UNIX client systems running the requester. If you use a host name on the HLAPI/2 or HLAPI/NT or HLAPI/USS client systems, the host name must be resolvable.