Note

Before using this information and the product it supports, be sure to read the general information under “Notices” on page v.

First Edition (October 1996)

This edition applies to Version 1 of Intelligent Decision Server, 5639–IDS, and to any subsequent releases until otherwise indicated in new editions or technical newsletters. Make sure you are using the correct edition for the level of the product.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address given below.

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Contents

Notices v
Trademarks v

About This Book vii
Who Should Read This Book vii
We'd Like Your Comments viii

Chapter 1. Introducing IDS Host Services 1
Accessing DB2 and CAS Applications from IDS Workstations 1
Examining Communications Between Data Access Components 4
Viewing DB2 Data Access Components 5
Viewing CAS Data Access Components 6
Understanding SNA APPC Communications Between Host and LAN 7
Identifying Host Services Address Spaces and Their Logical Connections . . . 9

Chapter 2. Preparing for DB2 Access 13
Configuring DB2 Authorization IDs 13
Creating Application Plans 14
Creating an IDS Database and Installing a WTDD 15
Upgrading a DIS Version 1.3.x WTDD 17
Optionally Defining Synonyms for New Users 18
Accessing a DB2 Database as a User 18
Creating Synonyms with Macros 21

Chapter 3. Using Host Console Commands 23
Summary List of Host Console Commands 23
Syntax Conventions 24
Command Reference 25
!DISABLE — Disable Service Address Space(s) 25
!DISPLAY — Display Configuration and Status Information 27
!ENABLE — Enable Service Address Space(s) 30
!SET — Change Configuration Setting 32
!SNAP — Capture Selected Data in a "Snapshot" 32
!START — Start Service Manager or Services for a Gateway 35
!STOP — Stop Services for a Gateway 39
!TRACE — Write Trace Information to SYSOUT 41
!WRITELOG — Control Logging of Console Messages 43

Chapter 4. Handling Messages 45
Identifying Log and Trace Messages Generated by Host Services 45
Contacting Support 46

Appendix A. Configuring IDS Data Access Tools to Use DRDA Servers . . 47

Appendix B. Configuration Parameter Reference 49

Sample Configuration Members 49
Control Member Parameter Reference 50
Gateway Communications Parameter Reference 56

Appendix C. Using the SQLMACRO Facility 59

Appendix D. Handling Messages 61
Identifying Log and Trace Messages Generated by Host Services 61
Contacting Support 62

Glossary 63

IDS Publications 71

Index 73

Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Subject to IBM's valid intellectual property or other legally protectable rights, any functionally equivalent product, program, or service may be used instead of the IBM product, program, or service. The evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, are the responsibility of the user.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents.You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
500 Columbus Avenue
Thornwood, NY 10594
U.S.A.

Licensees of this program who wish to have information about it for the purpose of enabling (1) the exchange of information between independently created programs and other programs (including this one) and (2) the mutual use of the information that has been exchanged, should contact:

IBM Corporation
555 Bailey Avenue, W92/H3
P.O. Box 49023
San Jose, CA 95161–9023

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

Trademarks

The following terms are trademarks of IBM Corporation in the United States or other countries or both:

ACF/VTAM	DXT	PROFS
AIX	Enterprise System/9000	PS/2
Application System/400	ES/9000	QMF
APPN	IBM	RACF
AS/400	MVS/ESA	RISC System/6000
Capsule	MVS/XA	RS/6000
DATABASE 2	OfficeVision	S/390

DataJoiner	OfficeVision/VM	SQL/DS
DB2	Operating System/2	System/360
DB2 for OS	Operating System/400	System/370
DB2/600	OS/2	Virtual Machine/Enterprise Systems Architecture
DFSORT	OS/400	VM/ESA
Distributed Relational Database Architecture	Personal System 2	VTAM
DRDA	Presentation Manager	Win-OS/2

C-bus is a trademark of Corollary, Inc.

PC Direct is a trademark of Ziff Communications Company and is used by IBM Corporation under license.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

Microsoft, Windows, and the Windows 95 logo are trademarks or registered trademarks of Microsoft Corporation.

Other company, product, and service names, which may be denoted by a double asterisk (**), may be trademarks or service marks of others.

About This Book

This book describes routine host administration tasks for IBM's Intelligent Decision Server (IDS) **Host Services** product in a Multiple Virtual Storage/Enterprise Systems Architecture (MVS/ESA) operating system environment. These tasks enable IDS local area network (LAN) workstation users to access IBM's DATABASE 2 (DB2) relational database management system (RDBMS); they also allow IDS workstation users to access host-resident legacy data and **cooperative applications** through IBM's Cooperative Application Services (CAS) facility.

Host Services is one member of the IDS family of products, which is composed of the following product categories:

Enterprise Services

Enterprise Tools

Data Access Services

Database Gateway Services

Host Services

IDS Access Services

Host Services supports the host side of a “bridge” that gives IDS LAN workstation users access to DB2 and CAS applications on an MVS host computer. The LAN side of this bridge is supported by an IDS OS/2 Database Gateway Service supporting DB2 sessions and MVS CAS sessions, and Enterprise Tools residing on workstation computers capable of running IBM Operating System/2 (OS/2) Version 2.1.

The Enterprise Tools category of IDS products offers several ways to directly access and analyze relational data for a variety of system environments. The data access capabilities are based on a client/server architecture, which allows the distribution of data on a variety of platforms that can be located remotely or attached directly to IDS networks. The client/server architecture enables all users to work from the same data.

Information in this publication is provided in a way that presumes that your organization has already installed and configured IDS Enterprise Services, Data Access Services, and Enterprise Tools software. Communications between the MVS host system and the database gateway service should be configured and enabled; and IDS LAN communications should be functioning properly.

Who Should Read This Book

This manual describes routine administrative tasks that are part of the duties of both the MVS systems programmer and the DB2 database administrator. It is intended for all administrators of the IBM Host Services program product who enable access to DB2 databases and MVS-host-resident CAS applications by IDS users. It is assumed that the reader has a working knowledge of:

MVS/ESA operations

TSO

The DB2 relational database management system

We'd Like Your Comments

Your feedback is important to us in providing you with the most accurate and high-quality information. If you have any comments about this book or any other IDS documenatation, please visit our Website at:

http://www.dss.ibm.com

There you'll find the Feedback Connection page where you can enter comments and send them to us.

Chapter 1. Introducing IDS Host Services

Host Services is a software component of the Data Access Services line of IBM Intelligent Decision Server (IDS) software products. Data Access Services encompasses a wide selection of software products that IDS users can combine to gain access to a broad assortment of applications and databases residing on various platforms. The Host Services product is a host-resident program that provides IDS workstation users with internetwork access to a variety of IBM host-based databases and applications.

This manual describes how to administer the Host Services product when it has been configured to support IDS local area network (LAN) workstation access to IBM DATABASE 2 (DB2) relational databases and/or **cooperative application** s residing on an MVS host computer. Access to host components of cooperative applications is accomplished through Host Services and IDS **Cooperative Application Services** (CAS).

Cooperative Application Services is a distributed IDS facility that allows IDS LAN workstation users to run command scripts that access applications and legacy databases on any MVS host system supported by IDS Host Services. Workstation users can perform powerful and complex analyses available to the host system, using all the data accessible to IDS and/or host systems. The CAS product is composed of software that resides partially on an MVS host system and partially on personal computers (PCs) attached to an IDS LAN.

Host Services runs in the MVS environment on IBM Enterprise System/9000 (ES/9000) (or equivalent) computers running the Multiple Virtual Storage/Enterprise Systems Architecture (MVS/ESA) operating system.

For information regarding Host Services planning, installation, and configuration in an MVS environment, refer to the following IDS publications:

Planning MVS Host-to-LAN Connectivity

Installing Host Services and Configuring Host-to-LAN Communications for MVS

Accessing DB2 and CAS Applications from IDS Workstations

IDS provides a uniform graphical user interface and a standard set of software tools to help LAN workstation users access a variety of relational database management systems (RDBMSs), including DB2. The type of RDBMS (DB2 or some other database product) is transparent to the workstation user. The user's view is of a data access tool having direct access to the tables of the database.

IDS also provides a **CAS transformer** tool to allow LAN workstation users to use command script languages like IBM's Restructured Extended Executor language (REXX) to run jobs or retrieve data from nearly any application, non-relational DBMS, or RDBMS on the host system. The range of host-resident software that a CAS transformer and its associated script can invoke is extensive. For example, application developers can create scripts that use flat files, DB2, Query Management Facility (QMF), and

C. For more information regarding application development for CAS, refer to the IDS publication *Developing Applications with Cooperative Application Services*.

This wide access to data and applications residing on the MVS host system is accomplished primarily through three IDS components:

Enterprise Tools software running on IDS LAN workstations

Database Gateway Service s software residing on an IDS-supported personal computer attached to an IDS LAN and configured for a specified number of DB2 and/or CAS **sessions**, or connections, with an MVS host computer

Host Services software running on an MVS host computer

The IDS OS/2 Database Gateway Service is an intermediary between the MVS host and IDS LAN workstations. One of several types of IDS database gateway services, it resides on a computer configured to run IBM Operating System/2 (OS/2) Version 2.1. It converts the information flow by repackaging data generated by an IDS LAN workstation from a LAN protocol into a form acceptable by the host system. The database gateway service also converts the information flow for data flowing from the host system to IDS LAN workstations. In addition, the database gateway service establishes access paths to host services. It manages access paths as a pool and assigns them for use through certain IDS workstation tools, including Query, Reporter, DBA, SQL Entry, Forms Design, and Data Entry. The OS/2 Database Gateway Service can accommodate several types of **gateways** supporting access to different data sources and running concurrently (for example, DB2, CAS, Operating System/400 (OS/400) and DATABASE 2 OS/2 (DB2 for OS)).

When access to a DB2 database or CAS application is requested by an IDS workstation, the workstation sends the request (for CAS applications, the workstation sends a REXX or command list (CLIST) procedure) to the database gateway service. The database gateway service communicates with the host through IBM Communications Manager/2, which resides on the same computer as the gateway service. Communications Manager/2 uses the Systems Network Architecture (SNA) Advanced Program-to-Program Communications (APPC) Logical Unit 6.2 (LU 6.2) protocol and is connected to the host through a LAN link or a Synchronous Data Link Control (SDLC) line for remote communications.

The gateway service uses Communications Manager/2 to pass SQL statements or data or CAS scripts to the appropriate **service address space** (host service) on the host. For DB2 requests, the receiving **SQL service address space** then passes the request to the appropriate DB2 subsystem; for CAS applications, the receiving **CAS service address space** executes the procedure through Time Sharing Option Extensions (TSO/E), invoking the host component of the cooperative application. Figure 1 on page 3 illustrates this flow.

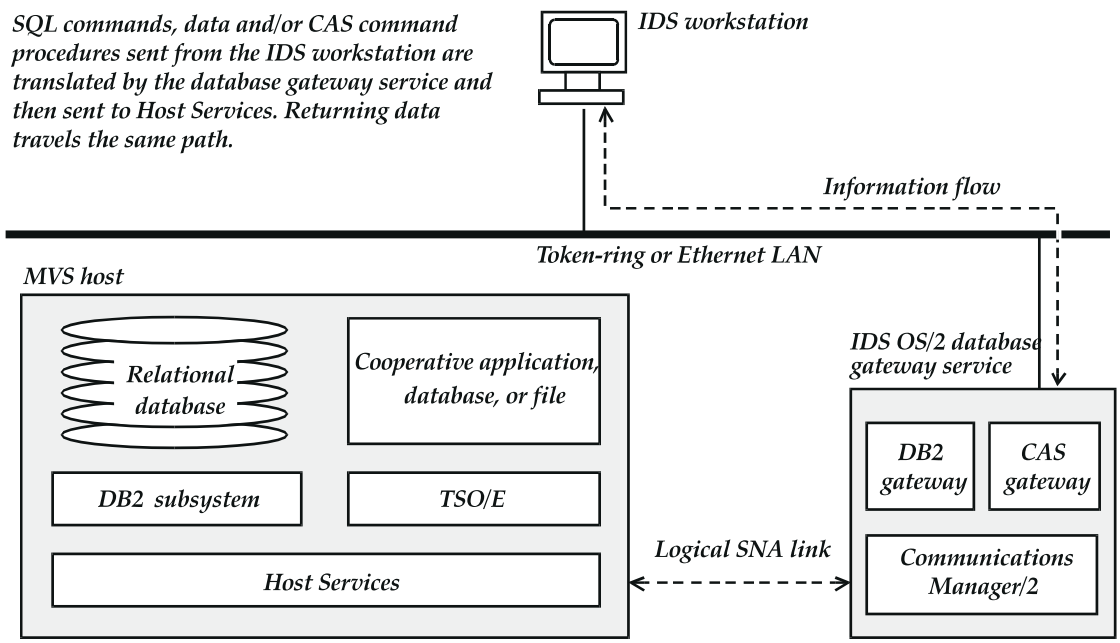


Figure 1. Access path from IDS workstation to DB2 and/or CAS application

If your DB2 subsystem provides Distributed Relational Database Architecture (DRDA) Requestor functions, then IDS data access tools can be configured to access the DRDA servers configured to that subsystem. For example, an IDS workstation user might access a Structured Query Language/Data System (SQL/DS) database residing on a remote virtual machine (VM) host system through a DRDA server configured to a DB2 subsystem residing on an MVS host connected to the IDS LAN. When using DRDA, IDS data access tools are configured to access the target database, and the secondary access path is transparent to users. See Appendix A, “Configuring IDS Data Access Tools to Use DRDA Servers” on page 47,” for more information about using DRDA. You can locate additional information about data access through DRDA in the IDS publication *Administering Databases with IDS*.

Host Services provides an MVS host interface interface that lets you administer the various Host Services address spaces and monitor and control sessions between the database gateway service and the host system. Detailed information about using the host interface command facility is provided in Chapter 3, “Using Host Console Commands” on page 23.

The database gateway service provides a interface interface that allows IDS administrators to enter gateway commands through an IDS **SA tool** (System Administration tool) from any IDS workstation. The set of gateway commands allows administrators to monitor and control sessions between IDS users' workstations and a host system.

For more information about the database gateway service and using it to manage DB2 and CAS sessions, refer to the IDS publications *Configuring and Administering Data-*

base Gateway Services and *System Administration Guide and Reference*. For information about installing a database gateway service on an IDS LAN, refer to the IDS publication *Installing IDS LAN Components*.

Examining Communications Between Data Access Components

IDS workstation tools are designed for direct communications with other LAN-resident services. Contrarily, DB2 is designed for direct access to applications running in MVS address spaces. Similarly, the CAS transformer runs on an IDS workstation, whereas the command script associated with the transformer runs in an MVS address space on the host. The widely distributed nature of these components requires several levels of communication to implement the simple view that the system presents to the IDS workstation user for either type of data access.

The IDS OS/2 Database Gateway Service uses several internal IDS processes common to both DB2 and CAS sessions to manage communications and convert data. Among them is a **monitor process** to manage allocation of internal gateway processes and resources.

A database gateway service can also use multiple **watchdog** processes, one for each DB2 or CAS gateway configured for the service. A watchdog process maintains APPC conversations with the host for commands and status, and it also manages cancel requests.

Associated with each watchdog process are a number of **session processes** used to perform data and format conversion. Each open data access tool on the developer's desktop is connected over the LAN to a session process. The session process performs bidirectional conversion between the network protocol and APPC protocol, and restarts the APPC connection if it is interrupted. One session process is active for each DB2 or CAS session between the gateway service and the host system. The number of sessions that each DB2 or CAS gateway is capable of running is determined by the IDS system administrator through the asset management process during installation.

The database gateway service uses Communications Manager/2 residing on the same computer to establish SNA APPC **sessions** with Host Services through Virtual Telecommunications Access Method (VTAM). You can choose the actual path between the VTAM address space and the gateway service from the several configurations that SNA allows to handle many possible connecting devices. Figure 3 on page 7 illustrates the SNA connection between Communications Manager/2 and VTAM.

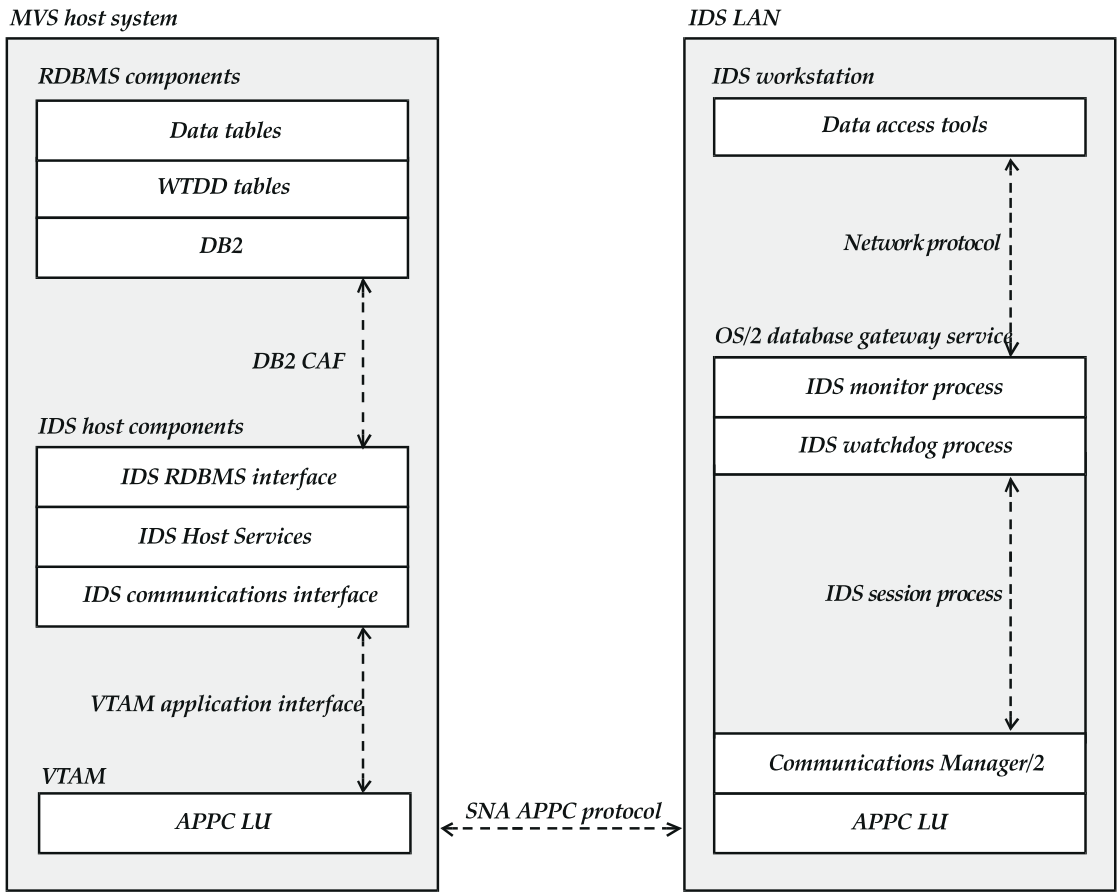


Figure 2. DB2 data access communications system view

At the host-system entry point into the SNA network, the IDS **communications interface** accesses the APPC communications protocol in sending and receiving messages between IDS Host Services and the communications network.

Viewing DB2 Data Access Components

When IDS workstation tools send data and SQL statements to the OS/2 Database Gateway Service, the database gateway service forwards them to the Host Services software and returns the results from Host Services to workstation tools for display or further manipulation. The database gateway service uses a token-ring or Ethernet LAN as the physical medium to communicate directly with data access tools used at IDS workstations. Figure 3 on page 7 shows the various DB2 data access components and their connections.

The Host Services components (called **host services** throughout this publication) reside in various MVS address spaces. The **service manager** component controls, maintains, and provides service administration capabilities for APPC conversations with

the watchdog process in the database gateway service. An **SQL service** component accepts DB2 SQL requests from the database gateway service and forwards them through the **IDS RDBMS interface** and the **DB2 Call Attach Facility (CAF)**. Each connection between Host Services and DB2 is established through CAF. When the DB2 subsystem returns results, Host Services passes them to the gateway service. See the "Identifying Host Services Address Spaces and Their Logical Connections" on page 9"section in this chapter for more information about host services address spaces used for DB2 access.

In Figure 3 on page 7, note the presence of the **Workstation Tools Data Dictionary (WTDD)** as a set of DB2 tables. The WTDD defines links between the data definitions in the **logical database** that the IDS workstation tools see and the actual data stored in the **physical database**. The data access tools use the WTDD to translate commands and data between the logical database and the physical database. Refer to *Administering Databases with IDS* for a more detailed description of the WTDD.

Viewing CAS Data Access Components

When a CAS transformer sends data and REXX or CLIST procedures to the database gateway service from an IDS capsule application running on a workstation, the database gateway service forwards them to the Host Services software on the MVS host system. The procedures invoke programs on the host system and they read and write host files according to REXX or CLIST commands executed through TSO/E. Figure 3 on page 7 shows the various CAS data access components and their connections.

The CAS host services reside in MVS address spaces. A **CAS service** invokes TSO/E to run the REXX or CLIST procedure supplied by the workstation. The CAS services share the same service manager with SQL services if SQL services are present. The service manager controls, maintains, and provides service administration capabilities for APPC conversations with the watchdog process in the gateway service. See the "Identifying Host Services Address Spaces and Their Logical Connections" on page 9" section in this chapter for more information about CAS host service address spaces.

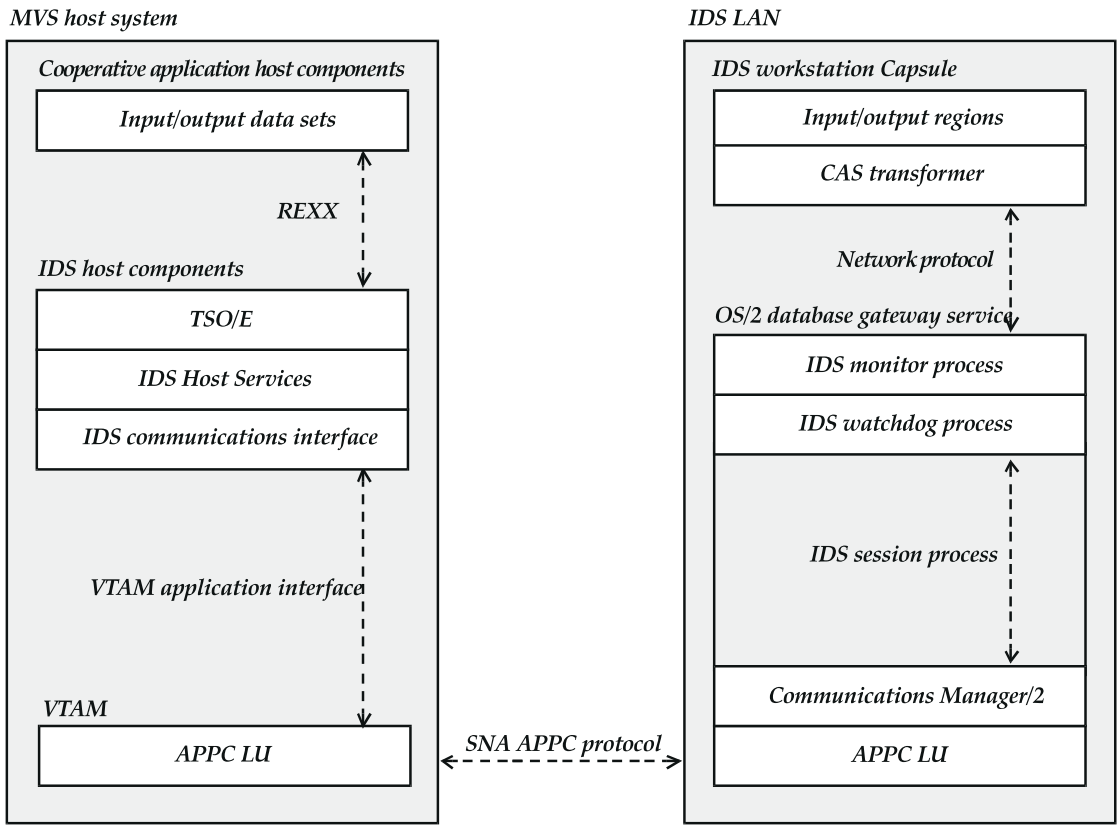


Figure 3. CAS communications system view

The REXX or CLIST procedures forwarded to Host Services address spaces invoke cooperative application programs on the host and read and write host files according to REXX or CLIST commands executed through TSO/E. A REXX procedure can return results to workstation tools from the host system for display or further manipulation. (A CLIST procedure must call a REXX procedure to return similar results from the host.)

Understanding SNA APPC Communications Between Host and LAN

The IDS OS/2 Database Gateway Service communicates with the host through IBM Communications Manager/2, which resides on the same computer as the database gateway service. Communications Manager/2 uses the SNA APPC LU 6.2 protocol and is connected to the host through a LAN link or an SDLC line for remote communications. In this topology, Communications Manager/2 participates directly as an SNA network **node**. In an SNA network, nodes are physical processors that are connected by physical data links.

At the heart of the SNA communications are **transaction programs** (TPs) on the host system and on the LAN. TPs are application programs that help users access resources in the network. Each TP runs within a particular **logical unit** (LU).

An LU is an addressable, logical processor within an SNA-defined network that lets a user gain access to network resources and communicate with other users. LU 6.2 is a particular type of SNA logical unit. LU 6.2 provides a connection between its users and network resources (TPs). The protocol that LU 6.2 provides is called Advanced Program-to-Program Communications (APPC). A TP uses the services provided by its LU to communicate with other TPs by issuing TP verbs (APPC functions). Figure 3 on page 7 and Figure 3 on page 7 show the APPC LUs that provide entry points into the SNA network for DB2 and CAS data access sessions.

The logical SNA connections between LUs are called **sessions**. Sessions can be compared to phone lines through which data flows between the LUs. An LU 6.2 can support more than one session concurrently with the same partner LU. Such sessions are called parallel sessions. An LU 6.2 can support sessions with multiple LUs.

While LUs are logically connected by sessions, TPs are logically connected by **conversations**. LU 6.2 treats a session as a reusable connection between two TPs. One session can support only one conversation at a time, but one session can support many conversations in sequence. TPs view only the conversation between them and are isolated from the underlying session that supports the conversation. A conversation uses an LU-LU session as a resource only for the duration of the conversation. After an LU-LU session has connected two Type 6.2 LUs, it remains active until a network operator or system process terminates it.

LU 6.2 does not allow an unlimited number of sessions between a pair of parallel-session capable LUs. **Session limits** are imposed on the number of total sessions LUs can have between them. A defined session limit also limits the number of conversations that can be active. If there are ten sessions, for example, then there can only be ten concurrent active conversations.

To establish a conversation, a TP specifies the name of the remote LU, the name of the TP, the **session mode name** (that identifies certain session characteristics like pacing levels and class of service), and security parameters. The LU name specified by the TP is a network addressable unit (NAU) that is used to route the connection within the SNA network.

If a session with the characteristics of the specified mode name exists between the two LUs and the session is not being used for another conversation, the LUs assign that session to the new conversation for its use. If a session is not available, the LU starts a new session using the specified mode name. The new session is then used for the conversation.

After the conversation is initiated, the two TPs use LU 6.2 verb functions to send and receive data as necessary to accomplish the transaction. When the transaction is finished, one TP ends the conversation. That session is now available for another conversation between TPs using the session's LUs as entry points into the SNA network.

For information about SNA APPC communications planning for Host Services in an MVS environment, refer to the IDS publication *Planning MVS Host-to-LAN Connectivity*.

Identifying Host Services Address Spaces and Their Logical Connections

An **address space** is a range of virtual storage addresses created and maintained by the MVS operating system for data and program residence and for program execution. Host Services address spaces and DB2 address spaces run in the same MVS system. Host Services uses three types of address spaces for managing DB2 and CAS processes and maintaining communications with the database gateway service:

SQL service address spaces

CAS service address spaces

Service manager address space

SQL Service Address Spaces

Host Services uses one SQL **service address space** for each connection to DB2 (each DB2 session). SQL services and DB2 operate in separate address spaces. The MVS host must have a separate set of service address spaces for each DB2 gateway, of which there can be several configured for a particular database gateway service. Each SQL service address space forwards database requests to DB2 and returns results from DB2, on behalf of a single instance of workstation tool use. Therefore, the number of service address spaces that you create in configuring Host Services for DB2 access directly limits the number of concurrent openings of data access tools configured to access DB2 on the IDS LAN.

Figure 4 on page 10 shows the communications path into a service address space. The database gateway service sends SQL statements and data to the service address space and receives data results on a single path, labeled **Commands and data** in Figure 4 on page 10.

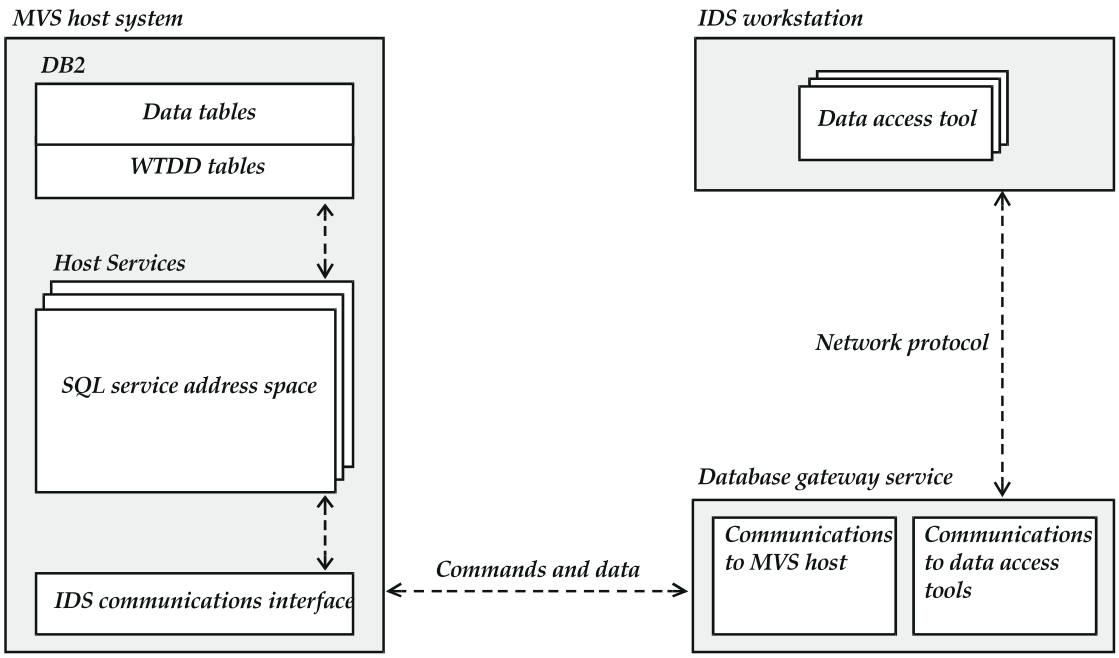


Figure 4. SQL service address space communications

The SQL service address space creates a CAF connection with DB2 when an icon is opened on the developer's desktop. The DB2 interface software in the service address space opens a thread for this connection when a workstation's **CONNECT** request is processed or an SQL logical unit of work (LUW) begins, depending on your configuration.

CAS Service Address Spaces

Host Services uses one **CAS service address space** for each connection to a CAS transformer (each CAS session). Each CAS service runs in a CAS service address space and invokes TSO/E to run the REXX or CLIST procedure supplied by the workstation. TSO/E and the REXX or CLIST procedure both execute in the same CAS service address space. The number of CAS service address spaces that you create in configuring Host Services directly limits the number of concurrent openings of CAS transformers on the IDS LAN.

Figure 5 on page 11 shows the communications path into a CAS service address space. The database gateway service exchanges data with the service address space through the IDS communications interface on a single path, labeled **Data** in Figure 5 on page 11.

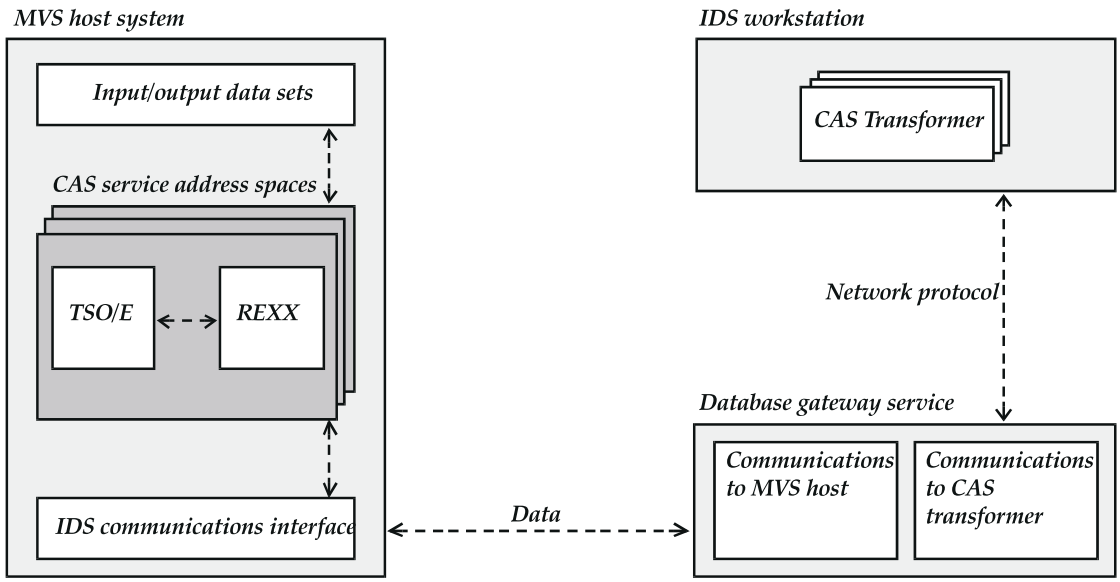


Figure 5. CAS service address space communications

Service Manager Address Space

Host Services also uses a single **service manager** address space to control the individual service address spaces. The MVS host must have a separate set of service address spaces for each gateway, but any number of gateways can share a single service manager address space. The **Host Services subsystem** provides a interface interface for operations.

Attention: Where this manual refers to "service manager," it means the service manager address space.

Figure 6 on page 12 shows two logical APPC communications paths, labeled **Host-to-gateway** and **Gateway-to-host**, between the service manager address space and each gateway. The gateway-to-host path carries commands from the gateway to the service manager, including commands to start or stop service address spaces. The gateway-to-host path also carries cancel requests to service address spaces through the service manager. The service manager uses the host-to-gateway path to report certain aspects of service address space status to the gateway, and to request selected functions for the gateway to perform.

There is no APPC path between the service manager and service address spaces; the Host Services subsystem schedules MVS **service request blocks** (SRBs) to carry operator commands issued from the host interface to the service manager, and to carry service address space logging messages from service address spaces to the service manager. While logging messages are passed across address spaces, tracing messages stay local to their service address spaces and are not passed to the service manager.

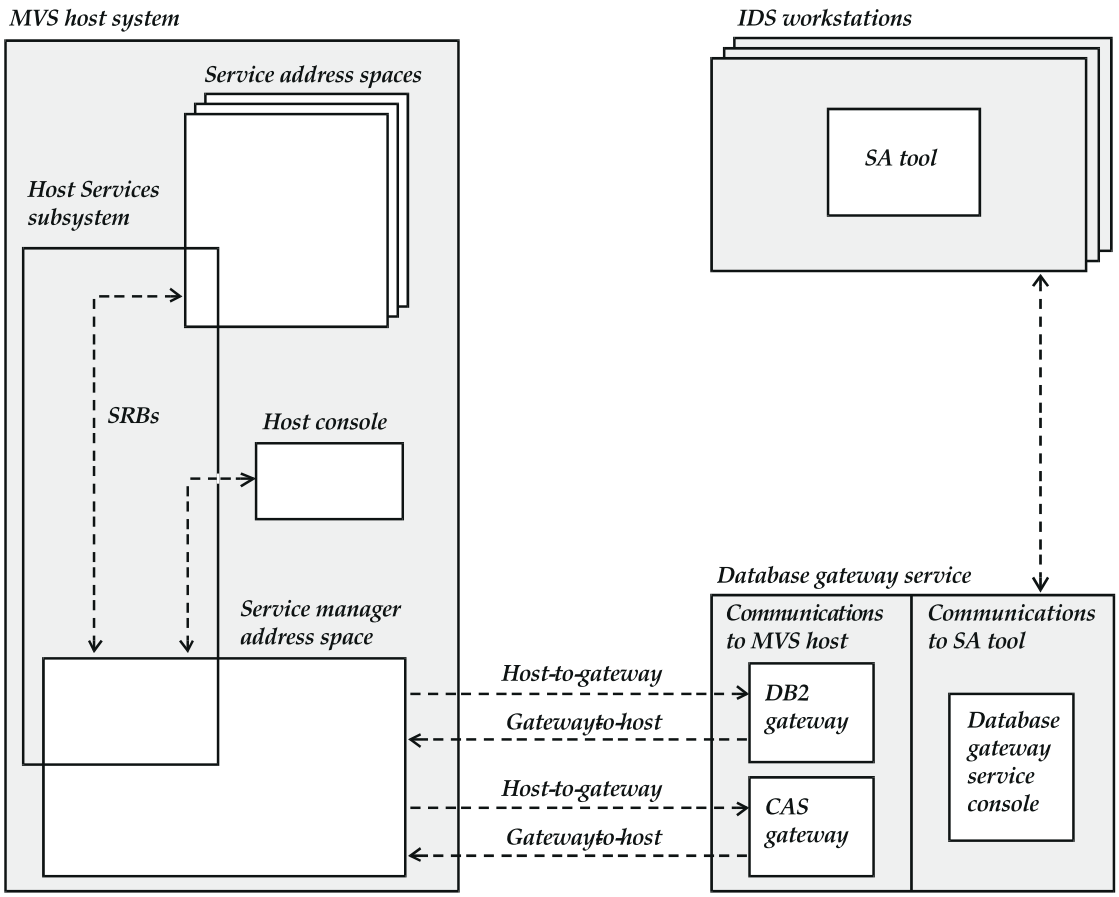


Figure 6. Service manager address space for administrative communications

Notice in Figure 6 that the database gateway service provides a interface interface that allows an IDS system administrator to enter gateway commands for the service manager through the SA tool. The IDS system administrator can use the SA tool through any IDS workstation, including the database gateway service machine if that computer has also been configured as an IDS workstation.

Chapter 2. Preparing for DB2 Access

This chapter provides information for the IDS database administrator (DBA) about preparing a DB2 subsystem for access through IDS workstation tools. As DBA, you must perform most of these activities to enable access to a new DB2 subsystem from Host Services. This chapter explains:

Configuring DB2 authorization IDs

Creating application plans

Creating an IDS database

Installing a new **Workstation Tools Data Dictionary** (WTDD)

Upgrading a DIS Version 1.3.**x** WTDD

Optionally defining synonyms for users

Completion of these tasks lets IDS workstation users begin accessing data in DB2 databases. The first two tasks can be performed only through the MVS host system; the remaining tasks can be performed either through the MVS host interface or through the SA tool on the IDS LAN. The first four tasks are required; the fifth task is required if you are migrating from a version of DIS to IDS; the last task is optional.

This chapter primarily describes DBA procedures that can be performed on the MVS host system rather than on the IDS LAN. IDS provides alternative database preparation methods so that you can choose the method that is most convenient for your organization. Refer to the IDS publication *Administering Databases with IDS* for complete information on preparing and maintaining database access for IDS workstation users.

Configuring DB2 Authorization IDs

You must have a DB2 authorization ID to build and control WTDDs. You must also have a DB2 authorization ID to bind the DB2 application plans used by Host Services. The authorization ID used in all examples in this manual is **METDBA**, but you can choose your own ID.

You must have **SELECT** privileges on the following DB2 catalog tables to use the IDS DBA tool to add entries to the WTDD that describe the physical database:

SYSIBM.SYSCOLUMNS

SYSIBM.SYSINDEXES

SYSIBM.SYSKEYS

SYSIBM.SYSSYNONYMS

SYSIBM.SYSTABLES

Before IDS users can access DB2 data, they must have DB2 authorization IDs. Some might already be time sharing option (TSO) users. Otherwise, you must define their user IDs to the host's security system.

Grant access to DB2 data selectively, as shown in the following example:

```
GRANT SELECT ON MKTDATA, ACME TO ALFRED, BETSY, CLARK, DEAN
GRANT ALL PRIVILEGES ON EMPLOYEE TO BETSY, DEAN
```

ALL PRIVILEGES does not include DBA rights; it simply grants all table privileges, such as SELECT, INSERT, DELETE, and others. You can grant only specific privileges, if you prefer.

Creating Application Plans

Each directly-connected DB2 subsystem (not using DDF or DRDA connections) requires a set of application plans. A set of Host Services application plans consists of two plans that are nearly identical, except that one is bound with the isolation level specified as “repeatable read,” and the other is bound with the isolation level specified as “cursor stability.” Each plan has the same root in its name: the root characters are followed by **R** for the plan using “repeatable read” or **C** for the plan using “cursor stability.” The plan name root is specified to Host Services through the **DB2PLAN** configuration parameter in the Control member. (See Table 3 on page 50.)

To create the application plans for a DB2 subsystem, edit and run the job **met .JCL(BIND)**. The JCL in **BIND** does the following:

- Binds the two application plans
- Grants execution privileges on the application plans to **PUBLIC**

Granting **EXECUTE** authority on the plans to **PUBLIC** incurs no security risk. The modules execute all SQL functions through dynamic SQL. Before performing each dynamic SQL statement, DB2 checks for proper authorization. Therefore, each workstation user can execute only those SQL statements for which that user is authorized.

If you choose not to grant **EXECUTE** privileges to **PUBLIC**, you must grant **EXECUTE** privileges individually for each IDS user of each application plan.

If you do not have IBM's **DSNTIAD** program available, then use **SPUFI** (through the ISPF **DB2I** interactive functions panel) to issue the SQL **GRANT** command found in the **BIND** job.

Minimum Privileges Required for Creating Application Plans:

- **BINDADD** for a new plan
- **BIND ON PLAN** for an existing plan

If you want to connect to DRDA servers, then you must alter the **BIND** commands in the **BIND** job to include packages. The underlying connections to the DRDA server must already be in place before you can successfully bind the packages.

The following example binds the packages and plans for the local DB2 system called **MVSDBP**, a remote DB2 system called **MVSDBT**, a remote SQL/DS system called

SQLDBA, and a remote Application System/400 (AS/400) system called **DB400**. The sample plan name root is **METPLAN**.

```
DSN SYSTEM(DSN)
BIND PACKAGE(MVSDBP.METPLANC)OWNER(METDBA)MEMBER(METSQL)ACTION(REPLACE)VALIDATE(BIND ISOLATION(CS)
        FLAG(I)RELEASE(COMMIT)
BIND PACKAGE(MVSDBT.METPLANC)OWNER(METDBA)MEMBER(METSQL)ACTION(REPLACE)VALIDATE(BIND ISOLATION(CS)
        FLAG(I)  RELEASE(COMMIT)
BIND PACKAGE(SQLDBA.METPLANC)OWNER(METPLANC)MEMBER(METSQL)ACTION(REPLACE)VALIDATE(BIND ISOLATION(CS)
        FLAG(I)RELEASE(COMMIT)
BIND PACKAGE(DB4  .METPLANC)OWNER(METPLANC)MEMBER(METSQL)ACTION(REPLACE)VALIDATE(BIND)ISOLATION(CS)
        FLAG(I)  RELEASE(COMMIT)
BIND PLAN(METPLANC)PKLIST( .METPLANC. )ACTION(REPLACE)VALIDATE(BIND)ISOLATION(CS)FLAG(I)RELEASE(COMMIT)
        OWNER(METDBA)

BIND PACKAGE(MVSDBP.METPLANR)OWNER(METDBA)MEMBER(METSQL)ACTION(REPLACE)VALIDATE(BIND ISOLATION(RR)
        FLAG(I)RELEASE(COMMIT)
BIND PACKAGE(MVSDBT.METPLANR)OWNER(METDBA)MEMBER(METSQL)ACTION(REPLACE)VALIDATE(BIND ISOLATION(RR)
        FLAG(I)  RELEASE(COMMIT)
BIND PACKAGE(SQLDBA.METPLANR)OWNER(METPLANR)MEMBER(METSQL)ACTION(REPLACE)VALIDATE(BIND ISOLATION(RR)
        FLAG(I)RELEASE(COMMIT)
BIND PACKAGE(DB4  .METPLANR)OWNER(METPLANR)MEMBER(METSQL)ACTION(REPLACE)VALIDATE(BIND ISOLATION(RR)
        FLAG(I)RELEASE(COMMIT)
BIND PLAN(METPLANR)PKLIST( .METPLANR. )ACTION(REPLACE)VALIDATE(BIND)ISOLATION(RR)FLAG(I)
        RELEASE(COMMIT)OWNER(METDBA)
END
```

After successfully running the **BIND** job, you can prepare to install a WTDD as described in the next section.

Creating an IDS Database and Installing a WTDD

This section describes how to install an IDS database and WTDD to enable workstation access to the end-user database tables in a DB2 subsystem. After you create an IDS database in a DB2 subsystem, you might want to create more than one WTDD in that database; or you might want to create an individual IDS database for each WTDD. The decision is up to you as the DBA.

Use the procedure in this section to perform the following required tasks:

- Designate or create an IDS database administrator
- Create an IDS database in a DB2 subsystem
- Install a WTDD in the IDS database
- Load a WTDD with initial DB2 catalog information

A WTDD is a logical data dictionary (a set of DB2 tables) that IDS uses to map data definitions in the **logical database** used by IDS workstation tools and the actual data stored in the **physical database**. By installing the WTDD, you provide database schema information to the workstation data access tools. You might choose to install more than one WTDD in a DB2 subsystem. For example, each department can have its own WTDD. A WTDD must be built in each DB2 subsystem to which an IDS user is allowed to connect. Refer to *Administering Databases with IDS* for a detailed description of the WTDD tables.

Before starting the WTDD installation process, you must be able to answer the following questions:

Who will own the WTDD tables? (**METDBA** is suggested.) The owner must be a user with **CREATEDBA** privilege.

What database and table space in the DB2 subsystem will contain the WTDD tables? The **WTCRTDB** job creates the database and storage groups.

Carrying out the procedure in this section requires the presence of the **met .SQLMACRO** data set. Substitute the data set name qualifier used in the installation process for the variable **met**. This data set is created by the **LOAD** job run from **met .JCL**. In addition, you need the **METWTBLD** cataloged procedure; it should have been created by the installation process. Finally, the IDS application plans must have been created as described in the preceding section.

Perform the following steps to create an IDS database and install a WTDD. Note the minimum DB2 privileges (if any) required for each step before proceeding.

1. Designate or create the IDS database administrator's user ID. **METDBA** is the user ID imbedded in the jobs supplied in **dis .JCL**. **METDBA** is shown as the user ID in all examples in this manual, but you can choose your own ID and password. Interpret **METDBA** to mean the user ID chosen for the IDS database administrator.

   Create **METDBA** if it does not already exist. **(METDBA** is required for step 4.)

2. **EDIT met .JCL(METWTBLD)** and copy it to a subsystem **PROCLIB** if this has not yet been done. This procedure executes the utility used in the following steps.

   Review the parameters at the start of the procedure and update them appropriately. Note that the **DB2PLAN** parameter specifies the name of the Host Services application plan that was bound with isolation level equal to “cursor stability.”

3. **EDIT met .JCL(WTCRTDB)**. You should perform this step for each DB2 subsystem that will contain a WTDD. This job does the following:

   Creates the specified **STOGROUPs** to be used by the database

   Creates the IDS database

   Grants **DBADM** and **USE OF BUFFERPOOL BP0** privileges on the database to the IDS database administrator

   Review the comments at the start of the job and update the JCL appropriately. Verify or update the variables as necessary in the **SYSIN** data set. Submit the job and verify that it has executed successfully.

   Minimum Privileges Required:

   CREATESG and **CREATEDBC**

   USE OF BP0 WITH GRANT OPTION (If the storage groups to be used by the database already exist then **USE OF stogroup WITH GRANT OPTION** is required.)

4. **EDIT met .JCL(WTLOAD)**. You should perform this step for each WTDD to be created in each DB2 subsystem.

This job creates the table space that will contain the WTDD tables and then creates the tables. It also loads initial table descriptions into the WTDD. The job loads the following descriptions by default:

A self description of the WTDD

A description of the tables in the DB2 catalog and communications databases

A description of the DB2 sample tables

A description of the DB2 **EXPLAIN** table

If you do not want to load the optional descriptions into the WTDD, remove the appropriate keywords from the **CTL** parameter. Review the comments at the start of the job and update the JCL appropriately. Verify/update the variables as necessary in the **SYSIN** data set. Submit the job and verify that it has executed successfully.

Minimum Privileges Required:

DBADM on the IDS database (granted in step 3)

5. **EDIT met .JCL(WTREORG)**. This job runs the DB2 **REORG** and **RUNSTATS** utilities against the WTDD table space.

   Review the comments at the start of the job and update the JCL and **SYSIN** data set appropriately. Submit the job and verify that it has executed successfully.

   Minimum Privileges Required:

   REORG

   STATS

Upgrading a DIS Version 1.3.x WTDD

The structure of the IDS WTDD differs from earlier DIS versions as follows:

Table **WT_COLUMNS** has a new column named **SQLNAME.**

Table **WT_TABLES** has a new column named **SQLNAME.**

Table **WT_VERSIONS** has two new columns: **WT_VARS** and **DD_VERSION.**

Table **WT_VARS,** a new table, has been added.

The following procedure upgrades WTDDs created by DIS Version 1.3.**x** systems as follows. The procedure:

Alters existing tables to add new columns

Creates the new table

Initializes the new columns and table

Adds self-describing information about the new columns and table

To upgrade a WTDD created by DIS Version 1.3.**x**, perform the following steps:

1. **EDIT met .JCL(WTUPGRAD)**. It executes the **METWTBLD** cataloged procedure to run the SQL in **met .SQLMACRO(WTDDUPGR)**.
2. Update the JCL and the variables in the **SYSIN** data set to specify the WTDD that you want to upgrade. To locate the necessary information from the DB2 catalog, you can run the following query:

   ```
   SELECT CREATOR,NAME,DBNAME,TSNAME FROM SYSIBM.SYSTABLES WHERE NAME='WT_VERSIONS'
   ```

3. Submit the job once for each WTDD. (Running the job against a WTDD that has already been upgraded has no negative effect.)

Optionally Defining Synonyms for New Users

To enable a user to access DB2 tables using the IDS workstation tools, you can optionally define appropriate DB2 synonyms for that user. This section explains how users access DB2 from IDS workstation tools and then describes how to set up synonyms to make such access possible.

Synonyms allow SQL to be generated with unqualified table names, yet still have access to tables with owners other than the current SQL ID setting. The same function can be performed by a view, but synonyms require less overhead on the part of DB2.

There are two areas in IDS where synonyms might be useful. First, the name of the WTDD owner can be specified in an Icon Options window (see Figure 7 on page 19)for any data access tool. If this field is left blank (for example, the icon has been upgraded from an earlier release of DIS that does not support this field), then the tool-generated queries of the WTDD will have unqualified table names. In this case, unless you are the owner of the WTDD, you must have synonyms for the WTDD table names.

Second, you might want to specify unqualified physical table names in your WTDD. In this case the SQL that is generated by the tools to access the end user table will contain unqualified physical table names that must be resolved by DB2. One method is to use the **SQL ID** field of the Icon Options window to set the current SQL ID to the desired table owner name. An alternative is to use synonyms.

Accessing a DB2 Database as a User

When an IDS workstation tool accesses a DB2 database, it specifies a DB2 authorization ID and password. For most IDS data access tools, the user or DBA specifies the DB2 authorization ID (**user ID**) and password in the **Other Data** section of the Icon Options window as shown in Figure 7 on page 19.

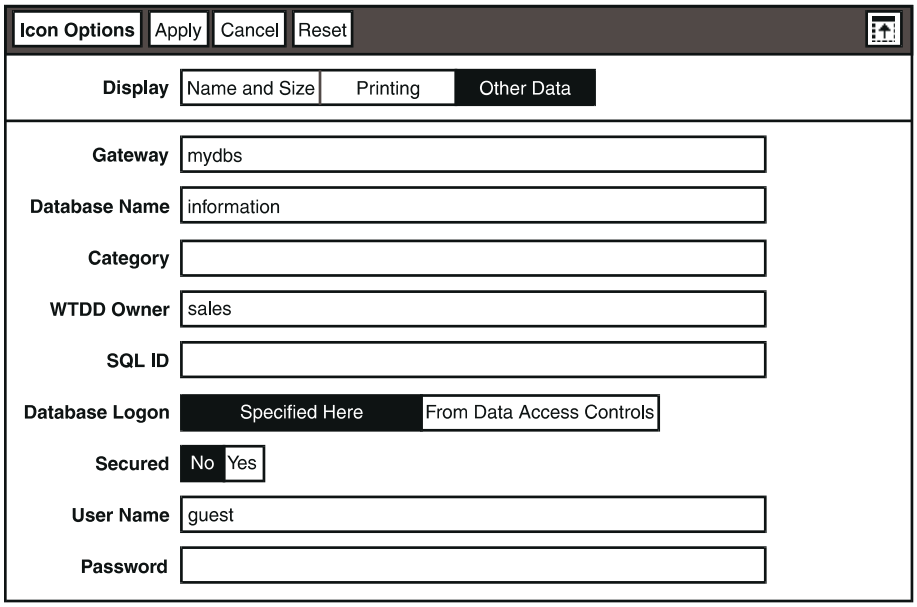


Figure 7. IDS data access tool Icon Options window

An authorized user or DBA uses the guidelines provided in the following table to enter the required information and select the appropriate choices in a data access tool's Icon Options window. If your server stores more than one physical database, you must identify the database you want to access in the **Database Name** field. It is unnecessary to supply a database name if only one physical database exists, or if you want to access a default database.

Field	Guidelines
Gateway	Type the name of the database gateway that you want the tool to access. If the gateway service is on a different realm from the one you are on, immediately follow the gateway name with a colon and then the realm name. For example: mydbs:HQ
Database Name	For DB2 databases, type the name of the DB2 subsystem you want to access. If you leave this field blank, or enter **dbs**, you will be connected to the DB2 subsystem specified by the configuration parameter **DB2SS**. If you want to connect to a supported remote IBM database using Distributed Relational Database Architecture (DRDA), immediately follow the local database name with a period and then the name of the DRDA server through which the connection is being made. For example: DSN.TD1
Category	Type either the print name or internal name of the database category needed for this tool. The name you specify here appears in the window header of the category window. (This field does not appear for all data access tools.)

Field	Guidelines
WTDD Owner	If the data access tool is configured to a database that supports multiple WTDDs and qualified table names, type the name of the owner of the WTDD tables. This field makes it unnecessary to drop and re-create synonyms whenever you want to access a different WTDD within the same database. (This field does not appear for all data access tools.)
SQL ID	For DB2 databases, type the table qualifier name. If it is desirable for Host Services to automatically issue the SQL statement **SET CURRENT SQLID**, you can enter the SQL ID here.
Database Logon	Select **Specified Here** if you want database access to be based on information entered in the **User Name** and **Password** fields. Select **From Data Access Controls** if you want access to be based on information in the desktop's Data Access Controls window.
Secured	Specify **Yes** if you want the data access tool icon's **User Name** field to be set to the user ID of the desktop on which it resides. Refer to *Administering Databases with IDS* for information about securing an icon.
User Name	Type a user name the database can identify as valid. This may or may not be the same as your desktop user ID depending on how your database security is set up. When the data access tool connects to the RDBMS, this is the user name used to derive the DB2 primary authorization ID.
Password	Type the database user's password. This password may or may not be the same as your desktop password depending on how your database security is set up. For security, the password is never displayed.

DB2 uses the authorization ID (derived from the value in the Icon Options window's **User Name** field) to verify ownership and access rights for tables in the database. The workstation user can use the tool to access tables owned by the DB2 authorization ID specified in the tool's Icon Options window. These tables are usually referred to as being owned by the workstation user ID. However, when the workstation user ID and DB2 authorization ID are different, the DB2 authorization ID is the actual owner of the tables.

To use IDS data access tools, users need access rights to the following classes of tables:

WTDD tables

Tables belonging to other users

Workstation access to any table requires that the following conditions be met:

The table must be described in the WTDD (except through the SQL Entry tool and SA tool).

The user must have access rights to the WTDD and the data table, either by owning the table or by having access rights granted by the owner or DBA.

For any table not owned by the user, you can either use the **SQL ID** field, create a synonym for the table, or put the fully qualified table name into the WTDD table-name columns. If the WTDD has unqualified table names, an **SQL ID** field entry or

synonyms are necessary because SQL requests generated by workstation tools, using unqualified table names, could otherwise access the wrong tables.

Creating Synonyms with Macros

Synonyms for the WTDD tables can be defined for a workstation tools user. In addition, a user might wish to access DB2 **SYSTEM** tables and the **EXPLAIN** table. The **createsynall** macro defines synonyms for all of these tables, or the individual macros can be called explicitly.

Running **createsynall** can be part of the procedure for adding a new user who will access DB2. The**createsynall** macro calls the macros listed in Table 1.

Table 1. Macros called by the createsynall macro

Macro	Action
createsynwtdd	creates synonyms for the WTDD tables
createsyndbmscat	creates synonyms for the DB2 catalog tables
createsynexplain	creates synonyms for the DB2 **EXPLAIN** tables

The SQL statement **EXPLAIN** retrieves information about the performance of SQL statements. The information is stored in a table named **PLAN_TABLE**. This table must be owned by the user performing the **EXPLAIN** command. You can run the job **met .JCL(EXPLAIN)** or call the macro **createuserexplain** to create **PLAN_TABLE** for a user.

For other tables to be accessible to workstation users, the tables must first be described in the WTDD. Those users (actually the DB2 authorization IDs used in the tools) who are to access the tables must be granted **SELECT** privileges on the tables. Finally, if a table name in the WTDD is not the fully qualified name, you can define a table synonym for each of those users. The **createsyn** macro creates synonyms for users on specific tables.

For instructions on executing the **createsynall** or **createsyn** macros, refer to *Administering Databases with IDS*.

Chapter 3. Using Host Console Commands

The Host Services subsystem provides a interface interface for operations. The interface can be a dedicated MVS operator interface or a TSO application such as System Display and Search Facility (SDSF). You can issue operator commands from the interface to administer the service manager, service address spaces, and database gateway sessions. Commands issued for a DB2 gateway or CAS gateway are not processed on the MVS host, but are sent to the database gateway service on the LAN for execution.

This chapter contains:

A summary list of the Host Services host interface commands

A description of command syntax conventions used in this publication

A detailed description of each Host Services command, with examples

Summary List of Host Console Commands

Table 2 lists Host Services host interface commands in alphabetic order. The detailed command descriptions in the remainder of this chapter are also alphabetically sorted by command for quick reference.

Table 2 (Page 1 of 2). Host Services commands

Command	Abbreviation	Action
!DISABLE	(none)	Disables a specified gateway's service address space, range of service address spaces, or all service address spaces.
!DISPLAY	!D	Displays, on a host interface, information about the service manager address space, specified service address spaces, gateways, and configuration parameters.
!ENABLE	(none)	Enables a specified gateway's service address space, range of service address spaces, or all service address spaces.
!SET	!T	Temporarily changes a specified configuration parameter.
!SNAP	(none)	Produces a formatted “snapshot” of specified data areas. Should be used only under the direction of IBM Technical Support for problem resolution.
!START	!S	Starts the service manager address space or enables communications with the specified gateway. This command can also cause a specified gateway to start specific service address spaces.
!STOP	!P	Terminates the service manager address space or communications with the specified gateway. This command can also cause a specified gateway to stop specific service address spaces.

Table 2 (Page 2 of 2). Host Services commands

Command	Abbreviation	Action
!TRACE	(none)	Controls writing of trace information to a SYSOUT file. Should be used only under the direction of IBM Technical Support for problem resolution.
!WRITELOG	!W	Controls logging of normal run time information to a central file.

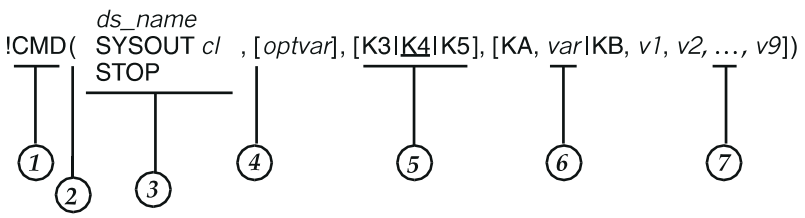
Syntax Conventions

You must begin each command with a **command recognition character**. The default command recognition character is an exclamation point (!), but you can specify a different character in SYS1.PARMLIB(IEFSSNxx). You can also use two characters in combination instead of using a single command recognition character. You can choose any non-alphanumeric character, such as @ or #, for the first command recognition character as long as it is not already in use by another subsystem. You have the option to specify any alphanumeric or non-alphanumeric character as the second command recognition character.

During configuration, you can include one or more Host Services commands in the Control configuration member met.CONFIG(METCNTRL). The Control configuration member specifies the default values for all of the service manager address space's configuration parameters. If you do include commands in the Control configuration member, you must precede each command with an exclamation point, even though your configuration might specify a different command recognition character. Refer to *Installing Host Services and Configuring Host-to-LAN Communications for MVS* for more information regarding MVS Host Services subsystem configuration and establishment of command recognition characters.

You can abbreviate most Host Services commands. Abbreviated forms are noted in Table 2 on page 23.

The following conventions describe command syntax in this publication.



1.	!CMD	An uppercase string of text indicates a command name or keyword. Type the text as shown. If your system's command interpreter is not case-sensitive, you can type commands and keywords in uppercase or lowercase. Text shown in lowercase, other than variables, must always be typed in lowercase. (The use of command recognition characters is described at the beginning of this section.)
2.	( )	Text enclosed in parentheses must be typed, including the parentheses.
3.	{ }	Text enclosed in braces is mandatory. Do not type the braces.
	• **ds_name fl** ˚ **SYSOUT cl** **Ë STOP**	Choices are sometimes shown stacked vertically within braces or brackets. If enclosed in braces, a stack has the same meaning as if all the choices were shown braced separately on the same line. One or more of the stacked choices must be typed. If enclosed in brackets, a stack has the same meaning as if all the choices were shown bracketed separately on the same line. Every choice shown within a bracketed stack is optional. Type choices in the order indicated, unless noted otherwise.
4.	[ ]	Text enclosed in brackets is optional. Do not type the brackets.
5.	**K3 | K4 | K5**	Vertical bars separate mutually exclusive choices within braces or brackets. Type no more than one choice, and do not type the vertical bars, unless directed to do so. (Some non-MVS IDS entry fields allow the vertical bar to be entered as a delimiter in path names.) An underlined choice indicates the default value.
6.	**var**	Variables are shown in slanted type. Substitute your own value for each variable.
7.	**...**	An ellipsis shown at the end of a vertical list of choices shown in braces or brackets indicates that you can type more than one of the choices shown vertically in the braces or brackets.

Command Reference

The remainder of this chapter describes each of the host interface commands in detail, including command syntax and usage examples. Commands are presented in alphabetic order.

!DISABLE — Disable Service Address Space(s)

This command disables a specified gateway's service address space, a range of service address spaces, or all service address spaces. The service address space does not terminate; currently connected users are allowed to complete their work.

Syntax

```
!DISABLE {gateway_name | gateway_identifier } {service_number
                [:service_number ] | ALL}
```

where:

gateway_name	identifies the configuration file member name for the gateway for which one or more service address spaces are to be disabled. For a list of valid gateway names, use the !DISPLAY GATEWAYS command.
gateway_identifier	specifies the 3-character identifier of the gateway for which one or more service address spaces are to be disabled. For a list of valid gateway identifiers, use the !DISPLAY GATEWAYS command.
service_number	identifies the service address space to be disabled. To disable a range of service address spaces, type the first service number of the range, followed by a space or colon (:), followed by the last service number of the range. To see a list of valid service numbers, use the !DISPLAY SERVICES command. A service number is composed of the last three characters of the service name displayed through the !DISPLAY SERVICES command. For example, the !DISPLAY SERVICES command might show a service name like this: ID=SQGWA002 In the example shown above, the last three characters, 002, comprise the service number. You can exclude leading zeros when entering a service number.
ALL	specifies that you want to disable all service address spaces for the specified gateway.

Remarks

The comma (,) and colon (:) are available as separators for your convenience as shown in the examples that follow. You can substitute a space for each of these characters.

Examples

The following examples disable the SQL service address space SQGWA001 controlled by the gateway GWASQS.

```
!DISABLE GWASQS, 1
+MCM2 59I! METOPER IDS Host Services DISABLE command executed.
!DISABLE GWA, 1
+MCM2 59I! METOPER IDS Host Services DISABLE command executed.
!DISABLE GWA,1
+MCM2 59I! METOPER IDS Host Services DISABLE command executed.
```

The following examples disable the range of CAS service address spaces between CAGWB001 and CAGWB004.

```
!DISABLE GWBCAS,  1:  4
+MCM2 59I! METOPER IDS Host Services DISABLE command executed.
!DISABLE GWB,1:4
+MCM2 59I! METOPER IDS Host Services DISABLE command executed.
!DISABLE GWB 1:4
+MCM2 59I! METOPER IDS Host Services DISABLE command executed.
!DISABLE GWB 1 4
+MCM2 59I! METOPER IDS Host Services DISABLE command executed.
```

The following example disables all SQL service address spaces for gateway GWASQS.

```
!DISABLE GWASQS ALL
+MCM2 59I! METOPER IDS Host Services DISABLE command executed.
```

!DISPLAY — Display Configuration and Status Information

This command displays, on a host interface, information about the service manager address space, specified service address spaces, gateways, and configuration parameters.

Syntax

{!DISPLAY | !D}

- CAS
- CONNECTED
- *gateway_identifier* [*service_number*[:*service_number*] | ALL]
- *gateway_name* [*service_number*[:*service_number*] | ALL]
- GATEWAYS
- OPTIONS
- PLANS
- *service_name*
- SERVICES
- SQS
- ...

where:

CAS displays the same information as !DISPLAY SERVICES, but only for CAS service address spaces.

CONNECTED displays the same information as !DISPLAY SERVICES, but only for service address spaces that are currently connected to IDS workstations.

gateway_identifier displays information about each gateway session from the perspective of the gateway. The 3-character gateway identifier can be followed by a service number (indicating a service address space), or range of service numbers, or ALL for all service numbers. For a list of valid gateway identifiers, use the !DISPLAY GATEWAYS command.

service_number

identifies the service address space to be displayed. To display information for a range of service address spaces, type the first service number of the range, followed by a space or colon (:), followed by the last service number of the range.

To see a list of valid service numbers, use the !DISPLAY SERVICES command. A service number is composed of the last three characters of the service name displayed through the !DISPLAY SERVICES command. For example, the !DISPLAY SERVICES command might show a service name like this:

ID=SQGWA002

In the example shown above, the last three characters, 002, compose the service number. You can exclude leading zeros when entering a service number.

ALL

specifies that you want to display information for all service address spaces for the specified gateway.

gateway_name

displays the same information as !DISPLAY gateway_identifier for the specified configuration file member name. For a list of valid gateway names, use the !DISPLAY GATEWAYS command. The gateway name can be followed by a service number (indicating a service address space), or range of service numbers, or ALL for all service numbers. (See service_number and ALL descriptions for gateway_identifier.)

GATEWAYS

displays the following information for every gateway specified in the Control configuration member:

The gateway name as specified in a GATEWAY statement in the Control configuration member (not necessarily the same as the name in the gateway's own configuration on the LAN). Refer to *Installing Host Services and Configuring Host-to-LAN Communications for MVS* for information about the Control configuration member.

The gateway type: C for Cooperative Application Services or S for SQL (DB2)

The 3-character gateway identifier

The current status of the gateway

OPTIONS

displays current Host Services configuration parameter settings, as shown in the following examples. Refer to *Installing Host Services and Configuring Host-to-LAN Communications for MVS* for information about Host Services configuration options.

PLANS

displays the DB2 application plan prefix used by IDS for each DB2 subsystem.

service_name

displays the same information as SERVICES, but only for the specified service address space.

SERVICES	displays the following information for every service address space controlled by the service manager address space: ID: the service name ASID: the hexadecimal address space identifier GATEWAY: the 3-character identifier of the gateway connected to the host service LU: the host service logical unit (LU) name STATUS: the current status of the host service, including the connected authorization ID (if the host service is connected)
SQS	displays the same information as !DISPLAY SERVICES, but only for SQL service address spaces.

Remarks

You can enter as many operands (shown stacked in braces) as you want following the command name. Separate each operand with a space or comma. Host Services will execute instructions in the specified order.

Examples

The first example displays the status of all configured service address spaces.

```
!DISPLAY SERVICES
+MCM2 62M ! METOPER IDS Host Services SERVICE DISPLAY:
+MCM2162M ! METOPER ID=SQGWA  2 ASID=     41 GATEWAY=GWALU=SQSGWA 2     STATUS=STARTUP COMPLE
+MCM2162M ! METOPER ID=SQGWA  3 ASID=     3F GATEWAY=GWALU=SQSGWA 3     STATUS=CONNECTED TO A
+MCM2162M ! METOPER     ID=CAGWB  1 ASID=     4     GATEWAY=GWB     LU=CASGWB 1 STATUS=CONNECTED
+MCM2 59I ! METOPER IDS Host Services DISPLAY command executed.
```

The next example displays the status of every gateway specified in the Control configuration member. It currently shows two gateways, GWASQS (DB2 gateway) and GWBCAS (CAS gateway).

```
!D GATEWAYS
+MCM2 64M ! METOPER     IDS Host Services GATEWAY DISPLAY:
+MCM2 73M ! METOPER     NAME=GWBCAS     TYPE=C     ID=GWB     STATUS=STARTUP COMPLETE
+MCM2 73M ! METOPER     NAME=GWASQS     TYPE=S     ID=GWA     STATUS=STARTUP COMPLETE
+MCM2 59I ! METOPER     IDS Host Services D command executed.
```

The next example displays current Host Services configuration option settings.

```
!DISPLAY OPTIONS
+MCM2 13M ! METOPER     IDS HOST SERVICES V2.1.  CONFIGURATION OPTIONS:
+MCM2 13M ! METOPER     SVCNUMBER=228                TRACE-CLASS=A
+MCM2 13M ! METOPER     LOGPOOL=1 ,1                OPERPOOL=1 ,1
+MCM2 13M ! METOPER     SRBPOOL=1 ,1                TANPOOL=8,8
+MCM2 13M ! METOPER     CONLEVEL=I                  LOGLEVEL=I
+MCM2 13M ! METOPER     MESSAGE-CASE=M              LOGMODE=DISMODE
+MCM2 13M ! METOPER     APPLID-GTH=METGTH           APPLID-HTG=METHTG
+MCM2 13M ! METOPER     INTERNALTRACE=              MAXECSA=1
+MCM2 13M ! METOPER     ROUTCDE=2
+MCM2 13M ! METOPER     ATTACHINTERVAL=3            STATUSINTERVAL=9
+MCM2 13M ! METOPER     MAXSERVICES=                STATUSEXIT=NONE
+MCM2 13M ! METOPER     CONNECTEXIT=METCNCTX        REQUIRE-DPF=N
+MCM2 13M ! METOPER     PERM-CONNECT=NO             THREAD=C
+MCM2 13M ! METOPER     DB2SS=DSN                   SQLEXIT=METSQLX
+MCM2 13M ! METOPER     CASMEM=1  K                 SQSMEM=1  K
+MCM2 59I ! METOPER     IDS Host Services DISPLAY command executed.
```

The next example displays the status of every gateway specified in the Control configuration member, followed by a list of all service address spaces that are currently connected to IDS workstations.

```
!D GATEWAYS,CONNECTED
+MCM2 64M ! METOPER     IDS Host Services GATEWAY DISPLAY:
+MCM2 73M ! METOPER     NAME=GWBCAS     TYPE=C     ID=GWB     STATUS=STARTUP COMPLETE
+MCM2 73M ! METOPER     NAME=GWASQS     TYPE=S     ID=GWA     STATUS=STARTUP COMPLETE
+MCM2 62M ! METOPER     IDS Host Services CONNECTED DISPLAY:
+MCM2162M ! METOPER     ID=CAGWB  3 ASID=     3F GATEWAY=GWB LU=CASGWB 3 STATUS=CONNECTED TO ADAN
+MCM2162M ! METOPER     ID=SQGWA  1 ASID=     4  GATEWAY=GWA LU=SQSGWA 1 STATUS=CONNECTED TO BAKE
+MCM2 59I ! METOPER     IDS Host Services DISPLAY command executed.
```

!ENABLE — Enable Service Address Space(s)

This command enables (activates) a specified gateway's service address space, a range of service address spaces, or all service address spaces.

Syntax

```
!ENABLE {gateway_name | gateway_identifier } {service_number
                [:service_number ] | ALL}
```

where:

gateway_name — identifies the configuration file member name for the gateway for which one or more service address spaces are to be enabled. For a list of valid gateway names, use the !DISPLAY GATEWAYS command.

gateway_identifier — specifies the 3-character identifier of the gateway for which one or more service address spaces are to be enabled. For a list of valid gateway identifiers, use the !DISPLAY GATEWAYS command.

service_number — identifies the service address space to be enabled. To enable a range of service address spaces, type the first service number of the range, followed by a space or colon (:), followed by the last service number of the range.

To see a list of valid service numbers, use the !DISPLAY SERVICES command. A service number is composed of the last three characters of the service name displayed through the !DISPLAY SERVICES command. For example, the !DISPLAY SERVICES command might show an SQL service name like this:

ID=SQGWA002

In the example shown above, the last three characters, 002, comprise the service number. You can exclude leading zeros when entering a service number.

ALL — specifies that you want to enable all service address spaces for the specified gateway.

Remarks

The comma (,) and colon (:) are available as separators for your convenience as shown in the examples that follow. You can substitute a space for each of these characters.

Examples

The following examples show four ways to enable the SQL service address space SQGWA002 controlled by gateway GWASQS.

```
!ENABLE GWASQS, 2
+MCM2 59I ! METOPER IDS Host Services ENABLE command executed.
!ENABLE GWA, 2
+MCM2 59I ! METOPER IDS Host Services ENABLE command executed.
!ENABLE GWA,2
+MCM2 59I ! METOPER IDS Host Services ENABLE command executed.
!ENABLE GWA 2
+MCM2 59I ! METOPER IDS Host Services ENABLE command executed.
```

The following examples show four ways to enable the range of CAS service address spaces between CAGWB002 and CAGWB005.

```
!ENABLE GWBCAS, 2: 5
+MCM2 59I ! METOPER IDS Host Services ENABLE command executed.
!ENABLE GWB,2:5
+MCM2 59I ! METOPER IDS Host Services ENABLE command executed.
!ENABLE GWB 2:5
+MCM2 59I ! METOPER IDS Host Services ENABLE command executed.
!ENABLE GWB 2 5
+MCM2 59I ! METOPER IDS Host Services ENABLE command executed.
```

The following example enables all service address spaces for gateway GWASQS.

```
!ENABLE GWASQS ALL
+MCM2 59I ! METOPER IDS Host Services ENABLE command executed.
```

!SET — Change Configuration Setting

This command temporarily changes the specified configuration parameter. For a list of the current settings, use the!DISPLAY OPTIONS command.The setting is changed only for the duration of the current execution of the service manager. To permanently change the setting, you must edit the file met.CONFIG(METCNTRL).

Refer to Appendix B, “Configuration Parameter Reference” on page 49,” for information about specific configuration parameters and their default settings.

Syntax

{!SET | !T} configuration_option [=|:] value

where:

configuration_option	identifies the parameter that you want to change.
value	specifies the value to be assigned to that parameter.

Remarks

The equal sign (=) and colon (:) are available for your convenience. Neither affects the operation of this command.

Example

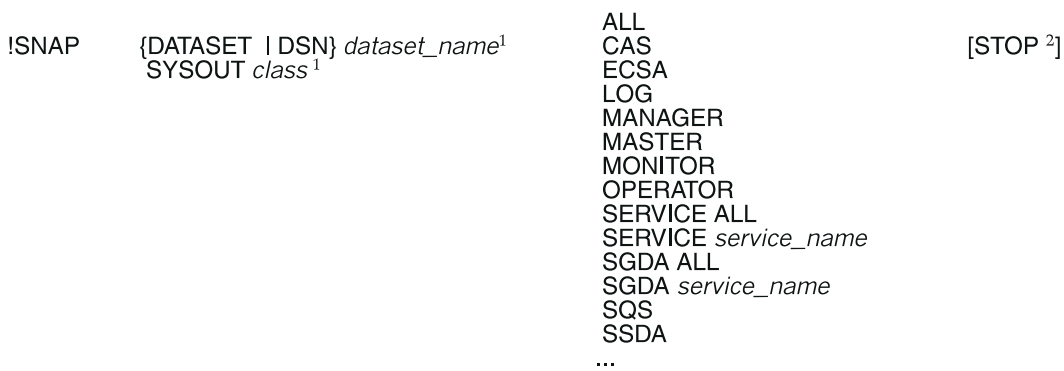
```
!SET CONLEVEL=E
```

!SNAP — Capture Selected Data in a "Snapshot”

This command produces a formatted “snapshot” of specified data areas.

Attention: Do not use the !SNAP command during normal operations. This command should be used only under the direction of IBM Technical Support for problem resolution.

Syntax



where:

ALL	captures all of the data areas listed below.
CAS	captures the global and local data areas for all CAS address spaces.
DATASET dataset_name	specifies the cataloged data set that is to contain the captured information. You can issue the abbreviated keyword DSN in place of DATASET.
ECSA	captures the Host Services data areas that are in the extended common service area (Extended CSA), which is an extension to the common service area (CSA) residing above the 16-megabyte line.
LOG	captures the data areas for the Log task.
MANAGER	captures the data areas for all service manager tasks.
MASTER	captures the data areas for the Master Control task.
MONITOR	captures the data areas for the Monitor task.
OPERATOR	captures the data areas for the Operator task.
SERVICE ALL	captures the global and local data areas for all Host Services (SQL and CAS) address spaces.

SERVICE service_name	captures the data areas for the specified service address space. For a list of active Host Services names, use the !DISPLAY SERVICES command.
SGDA ALL	captures the Service Global Data Areas (SGDAs) for all (SQL and CAS) service address spaces.
SGDA service_name	captures the SGDA for the specified service address space. For a list of active Host Services identifiers, use the !DISPLAY SERVICES command.
SQS	captures the global and local data areas for all SQL service address spaces.
SSDA	captures the Host Services Subsystem Data Area (SSDA).
SYSOUT class	specifies the SYSOUT class for the captured information.
STOP	closes and releases the data set opened previously with the !SNAP DATASET dataset_name or !SNAP SYSOUT command.

Notes

1. If used, this parameter must be specified first, before all other operands.
2. If used, this parameter must be specified last, following all other operands.

Remarks

You can list as many operands (shown stacked in braces in the syntax example) as you want following the !SNAP command name. Separate each operand with a space or comma. Host Services will execute instructions in the specified order.

To capture information successfully, you need to issue one or more!SNAP commands containing at least three operands specifying the following:

The destination of the captured information

The contents of the captured information

The STOP keyword to terminate the information capture

Examples

```
!SNAP SYSOUT X
!SNAP ALL
...
!SNAP STOP
```

The next example shows how you can list several operands following the !SNAP command name.

```
!SNAP SYSOUT A SSDA STOP
```

!START — Start Service Manager or Services for a Gateway

This command starts the specified service manager address space or starts communications with the specified DB2 or CAS gateway.

Syntax 1

```
{!START | !S} METSERV [,COLD] [control_member_name]
```

where:

METSERV	indicates that you want to start the service manager address space.
COLD	terminates any currently active service address spaces and reinitializes subsystem resources, before starting a new service manager address space. If you omit COLD, the startup does not affect existing subsystem resources such as service address spaces.
control_member_name	specifies the member containing the service manager configuration options to be used. If you omit control_member_name, the default is METCNTRL.

Remarks 1

Blanks are not allowed between parameters of this command.

When entered with the METSERV option (Syntax 1), this command starts the service manager address space. If a gateway is active but the service manager address space is not, the gateway repeatedly tries and fails to form its conversations with the service manager. When the service manager is started, the gateway can form its conversations. Then, the service manager automatically starts services in response to gateway requests.

Example 1

The following example starts a service manager address space named **DIST**.

```
!START METSERV
START METTCNTL.DIST,S=DIST,START=H,CTL=METCNTRL
ICH7 1I METTCNTL LAST ACCESS AT 11:44: 2 ON THURSDAY, FEBRUARY 25, 1993
$HASP1 METTCNTL ON STCINRDR
IEF693I PROCEDURE METTCNTL IS ASSIGNED TO USER METTCNTL, GROUP HOSTGRP
$HASP373 METTCNTL STARTED+CCM 1 1M ! METCNTRL Copyright 1984, 1994, IBM, Inc.
+CCM 1 2M ! METCNTRL All rights reserved.
+CCM 1 3M ! METCNTRL US Government Users Restricted Rights.
+CCM 1 4M ! METCNTRL Use, duplication or disclosure restricted
+CCM 1 5M ! METCNTRL By GSA ADP Schedule Contract with IBM Corp.
+MCM2 38I ! METCNTRL Gateway ID GWA has been successfully initialized.
+MCM2 38I ! METCNTRL Gateway ID GWB has been successfully initialized.
+MCM2 47M ! METCNTRL Successful COLD start of IDS Host Services.
+MCM2 43M ! METLOGTR IDS HOST SERVICES V2.1. INTERVAL STATUS REPORT:
+MCM2 43M ! METLOGTR ECSA IN USE=3224 MAX PERMITTED ECSA=65536
+MCM2 43M ! METLOGTR SRBBS IN USE = MAX SRBBS IN USE =
+MCM2 43M ! METLOGTR TANBS IN USE = MAX TANBS IN USE =
+MCM2 43M ! METLOGTR OCIBS IN USE = MAX OCIBS IN USE =
+MCM2 43M ! METLOGTR LTMBS IN USE = MAX LTMBS IN USE =
+MCM2 43M ! METLOGTR SGDAS IN USE = SGDAS ALLOCATED = 8
+MCM2 43M ! METLOGTR GWDAS IN USE = 2 GWDAS ALLOCATED = 2
```

Syntax 2

{!START | !S} {gateway_name }

where:

gateway_name — identifies the member name in the configuration file for the gateway that you want to start. This is the gateway name as specified in a GATEWAY statement in the Control configuration member, and not necessarily the same as the name in the gateway's own configuration on the LAN. For a list of valid gateway names, use the !DISPLAY GATEWAYS command.

Remarks 2

When entered with a gateway name (Syntax 2), the START command enables APPC conversations with the corresponding gateway. This allows Host Services to respond to subsequent requests from that gateway. If the gateway is already enabled, an error is reported.

Example 2

The following example starts communications between four host services (service address spaces) and the DB2 gateway (GWA) corresponding to gateway configuration member GWASQS. In this example, Service address SQGWA001 starts first, then services SQGWA002, SQGWA003, and SQGWA004.

```
!S   GWASQS
+MCM2 59I ! METOPER      IDS Host Services START command executed.
+MCM2 38I ! METCNTRL    Gateway ID GWA has been successfully initialized.
+MCM2131M ! METMONIT    Conversation established between TPN = XAHTGGWA and GATEWAY = GWA
+MCM2131M ! METCNTRL    Conversation established between TPN = XAGTHGWA and GATEWAY = GWA
+MCM2116I ! METCNTRL    Received START command for SERVICE = SQGWA  1.
START METTSQS.SQGWA  1,SSNM=DIST,LLU=DISGWA 1,RLU=LU4 1A  ,PX=1,LMOD=DISMODE,HCP= 37,TRC=N,SGD
$HASP1   METTSQS ON STCINRDR
IEF693I PROCEDURE METTSQS IS ASSIGNED TO USER ++++++++
$HASP373 METTSQS STARTED
+MCM2164I ! METMONIT      Service transaction program SQGWA  1 has started.
+MCM2131M ! SQGWA  1      Conversation established between TPN = SQGWA  1 and GATEWAY = GWA.
MCM2116I    ! METCNTRL      Received START command for SERVICE = SQGWA  2.
START METTSQS.SQGWA  2,SSNM=DIST,LLU=DISGWA 2,RLU=LU4 1A  ,PX=1,LMOD=DISMODE,HCP= 37,TRC=N,SGD
$HASP1   METTSQS ON STCINRDR
IEF693I PROCEDURE METTSQS IS ASSIGNED TO USER ++++++++
$HASP373 METTSQS STARTED
+MCM2164I ! METMONIT      Service transaction program SQGWA  2 has started.
+MCM2131M ! SQGWA  2      Conversation established between TPN = SQGWA  2 and GATEWAY = GWA.
MCM2116I    ! METCNTRL      Received START command for SERVICE = SQGWA  3.
START METTSQS.SQGWA  3,SSNM=DIST,LLU=DISGWA 3,RLU=LU4 1A  ,PX=1,LMOD=DISMODE,HCP= 37,TRC=N,SGD
$HASP1   METTSQS ON STCINRDR
IEF693I PROCEDURE METTSQS IS ASSIGNED TO USER ++++++++
$HASP373 METTSQS STARTED
+MCM2164I ! METMONIT      Service transaction program SQGWA  3 has started.
+MCM2131M ! SQGWA  3      Conversation established between TPN = SQGWA  3 and GATEWAY = GWA.
MCM2116I    ! METCNTRL      Received START command for SERVICE = SQGWA  4.
START METTSQS.SQGWA  4,SSNM=DIST,LLU=DISGWA 4,RLU=LU4 1A  ,PX=1,LMOD=DISMODE,HCP= 37,TRC=N,SGD
$HASP1   METTSQS ON STCINRDR
IEF693I PROCEDURE METTSQS IS ASSIGNED TO USER ++++++++
$HASP373 METTSQS STARTED
+MCM2164I ! METMONIT      Service transaction program SQGWA  4 has started.
+MCM2131M ! SQGWA  4      Conversation established between TPN = SQGWA  4 and GATEWAY = GWA.
```

Syntax 3

```
{!START | !S} {gateway_name | gateway_identifier } {service_number
                        [:service_number ] | ALL}
```

where:

gateway_name	start service address spaces for the specified gateway. This is the gateway name as specified in a GATEWAY statement in the Control configuration member, and not necessarily the same as the name in the gateway's own configuration on the LAN. For a list of active gateway names, use the !DISPLAY GATEWAYS command.
gateway_identifier	specifies the 3-character identifier of the gateway whose service address spaces are to be started.

service_number	identifies the service address space that you want to start. To start a range of service address spaces, type the first service number of the range, followed by a space or colon (:), followed by the last service number of the range. To see a list of valid service numbers, use the !DISPLAY SERVICES command. A service number is composed of the last three characters of the service name displayed through the !DISPLAY SERVICES command. For example, the !DISPLAY SERVICES command might show a service name like this: ID=SQGWA002 In the example shown above, the last three characters, 002, comprise the service number. You can exclude leading zeros when entering a service number.
ALL	starts all available service address spaces controlled by the specified gateway.

Remarks 3

When you enter the START command with a gateway identifier or gateway name in combination with a service identifier, range of service identifiers, or the keyword ALL, the gateway sends a command to the host causing the specified service address space(s) to start.

Example 3

```
!S   GWA,1:3
+MCM2116I ! METCNTRL      Received START command for SERVICE = SQGWA  1.
START METTSQS.SQGWA  1,SSNM=DIST,LLU=DISGWA 1,RLU=LU4 1A  ,PX=1,LMOD=DISMODE,HCP= 37,TRC=N,SGDA= 3
$HASP1   METTSQS ON STCINRDR
IEF693I PROCEDURE METTSQS IS ASSIGNED TO USER ++++++++
$HASP373 METTSQS STARTED
+MCM2164I ! METMONIT      Service transaction program SQGWA  1 has started.
+MCM2131M ! SQGWA  1      Conversation established between TPN = SQGWA  1 and GATEWAY = GWA.
MCM2116I ! METCNTRL      Received START command for SERVICE = SQGWA  2.
START METTSQS.SQGWA  2,SSNM=DIST,LLU=DISGWA 2,RLU=LU4 1A  ,PX=1,LMOD=DISMODE,HCP= 37,TRC=N,SGDA= 3
$HASP1   METTSQS ON STCINRDR
IEF693I PROCEDURE METTSQS IS ASSIGNED TO USER ++++++++
$HASP373 METTSQS STARTED
+MCM2164I ! METMONIT      Service transaction program SQGWA  2 has started.
+MCM2131M ! SQGWA  2      Conversation established between TPN = SQGWA  2 and GATEWAY = GWA.
MCM2116I ! METCNTRL      Received START command for SERVICE = SQGWA  3.
START METTSQS.SQGWA  3,SSNM=DIST,LLU=DISGWA 3,RLU=LU4 1A  ,PX=1,LMOD=DISMODE,HCP= 37,TRC=N,SGDA= 3
$HASP1   METTSQS ON STCINRDR
IEF693I PROCEDURE METTSQS IS ASSIGNED TO USER ++++++++
$HASP373 METTSQS STARTED
+MCM2164I ! METMONIT      Service transaction program SQGWA  3 has started.
+MCM2131M ! SQGWA  3      Conversation established between TPN = SQGWA  3 and GATEWAY = GWA.
```

!STOP — Stop Services for a Gateway

This command terminates the specified service manager address space or stops communications with the specified gateway.

When you enter this command with METSERV, the service manager address space does the following:

Shuts down all service address spaces

Shuts down all service manager subtasks

Frees global data areas in ECSA

Exits to MVS, causing the service manager address space to terminate

Syntax 1

{!STOP | !P}

ALL [NOPROMPT]
METSERV [NOPROMPT]
gateway_identifier
gateway_name
service_name

where:

ALL	specifies that you want to stop all service address spaces.
METSERV	specifies that you want to stop the service manager address space and all of its service address spaces. The gateway is automatically stopped with this command.
gateway_identifier	specifies the 3-character identifier of the gateway whose APPC conversations are to be stopped.
gateway_name	identifies and stops communications with the specified gateway. This is the gateway name as specified in a GATEWAY statement in the Control configuration member, and not necessarily the same as the name in the gateway's own configuration on the LAN. For a list of active gateway names, use the !DISPLAY GATEWAYS command.
NOPROMPT	avoids display of a message to require you to confirm or cancel the command.
service_name	identifies the service address space to be stopped. For a list of valid service names, use the !DISPLAY SERVICES command.

Remarks 1

When you enter the MVS CANCEL command with a service address space name, the specified host service is terminated, but the gateway tries to restart it within a few

seconds (unless you first enter the DISABLE or STOP gateway command to disable the host service).

Example 1

The following example terminates communications with gateway GWA.

```
!P   GWA
+MCM2 59I ! METOPER      IDS Host Services P command executed.
+MCM2161I ! METMONIT     Service transaction program SQGWA  3 has terminated.
+MCM2161I ! METMONIT     Service transaction program SQGWA  6 has terminated.
+MCM2161I ! METMONIT     Service transaction program
SQGWA  7 has terminated.
+MCM2161I ! METMONIT     Service transaction program SQGWA  4 has terminated.
+MCM2161I ! METMONIT     Service transaction program SQGWA  5 has terminated.
+MCM2161I ! METMONIT     Service transaction program SQGWA  2 has terminated.
+MCM21 2M ! METMONIT     XAHTGGWA deallocating conversation with gateway GWA.
+MCM2135M ! METCNTRL     Conversation outage between
TPN = XAHTGGWA and GATEWAY = GWA.
+MCM21 2M ! METMONIT     XAGTHGWA deallocating conversation with gateway GWA.
+MCM2135M ! METCNTRL     Conversation outage between
TPN = XAGTHGWA and GATEWAY = GWA.
+MCM2 18M ! METOPER      Communication with gateway GWA has been successfully terminated.
```

Syntax 2

```
{!STOP | !P} {gateway_name | gateway_identifier } {service_number
                          [:service_number ] | ALL}
```

where:

gateway_name	identifies and stops service address spaces controlled by the specified gateway. This is the gateway name as specified in a GATEWAY statement in the Control configuration member, and not necessarily the same as the name in the gateway's own configuration on the LAN. For a list of active gateway names, use the !DISPLAY GATEWAYS command.
gateway_identifier	specifies the 3-character identifier of the gateway whose service address spaces are to be stopped.

service_number — identifies the service address space that you want to stop. To stop a range of service address spaces, type the first service number of the range, followed by a space or colon (:), followed by the last service number of the range.

To see a list of valid service numbers, use the !DISPLAY SERVICES command. A service number is composed of the last three characters of the service name displayed through the !DISPLAY SERVICES command. For example, the !DISPLAY SERVICES command might show a service name like this:

ID=CAGWB002

In the example shown above, the last three characters, 002, constitute the service number. You can exclude leading zeros when entering a service number.

ALL — stops all service address spaces controlled by the specified gateway.

Remarks 2

When you enter this command with a gateway name or number in combination with a service number, range of service numbers, or the keyword ALL, the gateway sends a command to the host system causing the specified service address spaces to stop. This command prevents the gateway from automatically restarting the service address spaces.

Example 2

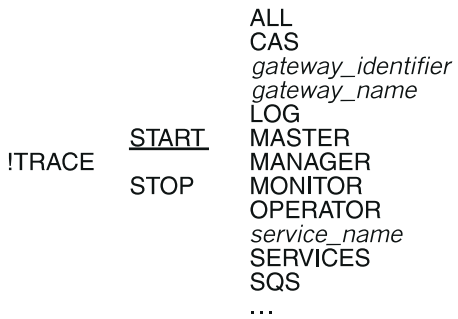
```
!STOP  GWA,3
+MCM2 59I ! METOPER     IDS Host Services P command executed.
+MCM2161I ! METMONIT    Service transaction program SQGWA  3 has terminated.
```

!TRACE — Write Trace Information to SYSOUT

This command lets you control writing of trace information to a SYSOUT file. The class of the SYSOUT file is taken from the TRACE-CLASS configuration parameter.

Attention: Do not use the !TRACE command during normal operations. This command should be used only under the direction of IBM Support for problem resolution.

Syntax



where:

ALL	activates all tracing listed below.
CAS	activates tracing in all CAS service address spaces.
gateway_identifier	activates tracing in all service address spaces belonging to the specified gateway.
gateway_name	activates tracing in all service address spaces belonging to the specified gateway.
LOG	activates tracing in the Log task.
MANAGER	activates tracing in all service manager tasks.
MASTER	activates tracing in the Master Control task.
MONITOR	activates tracing in the Monitor task.
OPERATOR	activates tracing in the Operator task.
service_name	activates tracing in the specified service address space. For a list of active Host Services names, use the !DISPLAY SERVICES command.
SERVICES	activates tracing in all service address spaces.
SQS	activates tracing in all SQL service address spaces.
START	commences tracing. This is the default setting assumed by Host Services if neither START nor STOP is explicitly entered to commence or terminate tracing.

STOP	terminates tracing. Closes and releases the data set opened previously with the !TRACE START command.

Remarks

You can list as many operands (shown stacked in braces) as you want following the !TRACE command name. Separate each operand with a space or comma. Host Services will execute instructions in the specified order.

Examples

The START keyword is implicit in the first line of the !TRACE example shown below.

```
!TRACE ALL
...
!TRACE STOP
```

The next example shows how you can list several operands following the !TRACE command name.

```
!TRACE START SQGWA  1, SQGWA  2
...
!TRACE STOP
```

!WRITELOG — Control Logging of Console Messages

This command lets you control logging of host interface messages to a central file.

Syntax

{!WRITELOG | !W} *dataset_name*
SYSOUT *class*
STOP

where:

dataset_name	specifies the cataloged data set that is to contain the logged information. If a log file is already open, that file is closed and logging continues in the newly specified one.
SYSOUT class	specifies the SYSOUT class for the logged information. If a log file is already open, that file is closed and logging continues in the newly specified one.
STOP	terminates logging of interface messages. Closes and releases the data set opened previously with the !WRITELOG dataset_name or !WRITELOG SYSOUT command.

Remarks

If you include a **SYSLOG DD** statement in the service manager cataloged procedure, logging starts automatically when you start the service manager address space.

Examples

```
!WRITELOG SYSOUT X
...
!WRITELOG STOP
```

Chapter 4. Handling Messages

Host Services generates status and error messages when necessary to alert you to problems or conditions that need attention. Some of these messages appear in only one place, a trace file. Others appear in both the trace file and the service manager log file. Others appear in both of these files and on the MVS host interface. The appearance of messages is controlled by the **CONLEVEL** and **LOGLEVEL** configuration parameters. Refer to Appendix B, "Configuration Parameter Reference" on page 49," and the IDS publication *Installing Host Services and Configuring Host-to-LAN Communications for MVS* for more information about these configuration parameters.

To interpret any status or error message, refer to the IBM IDS *Messages and Codes*. In *Messages and Codes*, words in italic type are variables representing names or numbers reported within the messages. For example, in the following message, **string** is the command that is in error:

94.236 13.14. MCM2 57M ! METOPER **string** is not a valid Host Services command

Identifying Log and Trace Messages Generated by Host Services

The *Messages and Codes* lists messages generated by Host Services that can appear on the MVS interface and in the log file and trace file. The log file is governed by the **!WRITELOG** command, as described in the preceding chapter. The trace file is governed by the **!TRACE** command, also described in the preceding chapter. The **CONLEVEL** configuration parameter controls which operations-related messages appear on the MVS interface. The **LOGLEVEL** configuration parameter controls which messages appear in the Host Services log file.

Each message is preceded by the date and time it was generated. The date is expressed as year and day of year, and the time is expressed in hours, minutes, and seconds, as shown in the following example:

94.236 13.14. MCM2 59I ! METOPER
IDS Host Services P command executed.

The sample message shown above was generated on the 236th day of 1994, at exactly 1:14 p.m.

The message itself begins with a message identifier comprising the Host Services component (**MCM**), four decimal digits (**2059** in the example), and a 1-character severity flag (**I** in the example).

The severity flag is **M**,**T, I**, **W**, **E**, or **S** :

M indicates an informational message that always appears on the interface, regardless of the **CONLEVEL** setting.

T indicates a trace message.

I indicates an informational message.

W identifies a warning message.

E identifies a recoverable error.

S indicates a severe error.

The exclamation mark (!) that follows the message number and severity flag in the preceding example indicates that the message was generated from the IDS Host Services subsystem whose subsystem command recognition character has been designated as "!".

Not all of the messages listed in *Messages and Codes* appear on the interface. The configuration file includes a **CONLEVEL** flag through which you indicate the lowest message severity to be displayed. For example, if you set the flag to **E**, the interface displays **S** and **E** messages but not **W** or**I** messages.

After the identifier, each message shows its origin. If the message came from the service manager address space, as in the previous example, it shows the name of the task that issued the message (**METOPER**, in this case). If the message came from a service address space, it shows the corresponding TP name (**SQGWA004**, for example).

In many cases, the most recent message is preceded by other messages that indicate the underlying error more specifically.

Contacting Support

In some cases, the problem solution requires that you contact your support organization. The solution will come more quickly if you follow these steps first:

1. If the error is reproducible and the log was not active: use the **WRITELOG** command to start the logging facility, and then reproduce the error.
2. Print a listing of the current log file and interface messages.
3. Show the log and interface listings to your support organization.

If your problem requires assistance beyond the capabilities of your own support group, consult the preface for detailed instructions on obtaining telephone assistance from IBM Technical Support.

Appendix A. Configuring IDS Data Access Tools to Use DRDA Servers

If your DB2 subsystem provides Distributed Relational Database Architecture (DRDA) Requestor functions, then you can access the DRDA servers configured to that subsystem in one of three ways:

1. In the **Database Name** field of the data access tool's Icon Options window, type the name of the local DB2 subsystem followed by a period and the DRDA server name in the format **local_database.DRDA_server**. For example:

   DSN.DB4

   If requested by the tool, the database type for the remote server is returned to the IDS workstation. For example, if a DBA tool connects through DB2 to an Operating System/400 (OS/400) database, when the DBA tool requests the database type, the response will be OS/400, not DB2.

2. In IDS tools that let you enter SQL statements, use the SQL **CONNECT** command to establish the connection:

   CONNECT TO **DRDA_server**

3. Use the IDS **CONNECT** extended SQL statement to establish the connection:

   CONNECT **user_ID** {/ | IDENTIFIED BY} **password** TO **database** [**.DRDA_server**]

   In this method of DRDA access, the syntax of the database name in the IDS **CONNECT** request includes a remote database name qualifier. It is separated from the local database name with a period. For example, to connect to a remote database named **DB400** from a local database named **DSN**, the Icon Options window would specify **DSN.DB400**.

For more information about configuring a specific IDS data access tool, refer to the IDS publication that describes the tool. You can locate additional information about data access through DRDA in the IDS publication *Administering Databases with IDS*.

See “Creating Application Plans” on page 14, in Chapter 2, “Preparing for DB2 Access” on page 13, for information about binding the IDS application plans using packages.

Appendix B. Configuration Parameter Reference

Host Services configuration information is specified within a partitioned data set whose DD name is **SYSPARM**. One member in this data set is the Control member. (The default member name is **METCNTRL**.) It contains configuration information applicable to the Host Services service manager address space, including information that is common to all gateways that communicate with that service manager address space. Separate gateway configuration members contain configuration information that is unique to the individual gateways. Configuring Host Services is described in detail in the IDS publication *Installing Host Services and Configuring Host-to-LAN Communications for MVS*.

Another way to specify configuration parameters is by using the Host Services command **!SET**. Parameters in the Control member are applied during initialization of the service manager address space. After initialization, if you use the **!SET** command, the new value applies until the service manager address space is stopped or the value is reset. Some configuration parameters might not take effect for currently running services; they will take effect when the service is restarted. The only way to reconfigure values in a gateway configuration member is to edit the appropriate member, then use the Host Services commands **!STOP** and **!START** from the MVS interface to stop and re-enable communications with the corresponding gateway.

The **!STOP gateway_name** command terminates all service address spaces that are associated with the specified gateway.

Sample Configuration Members

The sample configuration data set (**met .CONFIG**) distributed with the Host Services software contains three members:

METCNTRL, the Control member, holds default values for all configuration parameters along with **GATEWAY** statements to specify the gateway configuration members.

GWASQS is an SQL gateway configuration member describing DB2 gateway **GWA**.

GWBCAS is a CAS gateway configuration member describing CAS gateway **GWB**.

These members are listed with information about configuring Host Services in *Installing Host Services and Configuring Host-to-LAN Communications for MVS*.

Each **GATEWAY** statement in **METCNTRL** requires the presence of a separate gateway configuration member in the **met .CONFIG** data set, containing configuration information for that gateway. Some Host Services commands, such as **!START**, use the gateway configuration member's name.

The TP name associated with a service's APPC conversation is not explicitly configured. Instead, Host Services determines it during execution, using the following format:

ppgggnnn

where:

pp is the **CPREFIX** value specified in the gateway configuration member. This prefix is **SQ** for SQL gateways or **CA** for CAS gateways.

ggg is the gateway identifier, also specified in the gateway configuration member, indicating the gateway with which the Host Services address space is identified. All examples and discussions in this manual assume that the gateway identifier is **GWA** for SQL host services or **GWB** for CAS host services.

nnn is a 3-digit, decimal, Host Services service address space identifier number. The numbering (within the set of service address spaces associated with gateway **ggg**) starts with **001**.

A set of eight service address spaces for communicating with an SQL database (DB2) gateway might be named **SQGWA001**, **SQGWA002**, and so forth, through **SQGWA008**. For communicating with a CAS gateway, the service address spaces might be named **CAGWB001** through **CAGWB008**.

APPLID and LU names are site-selectable, but the mapping of TP names to LU names must be the same in the configuration of Host Services as in Communications Manager/2 residing with the gateway on the LAN.

The rest of this appendix describes the parameters in the Control member and in the gateway configuration member. The parameter names cannot be abbreviated.

Control Member Parameter Reference

Table 3 lists the parameter keywords in the Control member, together with each parameter's default value (if any), description, and legal values.

Table 3 (Page 1 of 7). Parameters in the Control member

Keyword	Default	Description
!any valid service manager command	—	Performs the command when the Host Services service manager address space is started. Syntax is as described in Chapter 3, “Using Host Console Commands” on page 23.
APPLID-FOR-GTH	—	Specifies the application ID name for the service manager task's APPC conversations with all of its gateways.
APPLID-FOR-HTG	—	Specifies the application ID name for the Monitor task's APPC conversations with all of its gateways.

Table 3 (Page 2 of 7). Parameters in the Control member

Keyword	Default	Description
ATTACHINTERVAL	**30**	Specifies the number of seconds to wait before timing out on completion of subtask **ATTACH**. If the interval expires, the requesting task terminates. If a Host Services address space cannot start its subtasks, it terminates its own initialization.
CASMEM	**100**	The number of kilobytes of dynamic storage to be used by CAS data encoding/decoding. Transferring complex data between the host and workstation might require more dynamic storage than provided by default.
CASPROC	**METCAS**	Specifies the name of the cataloged procedure that the service manager task uses to start a CAS service address space. Use of different Host Services procedures lets the **PERFORM** parameter (on the **EXEC** statement) specify different performance groups for different Host Services subsystems or gateways. It also permits use of alternate **STEPLIB** s for testing subsystems or gateways. You can specify this cataloged procedure in either the Control member or a gateway configuration member.
CONLEVEL	**W**	Specifies the minimum severity level for messages to be written to the MVS interface. In order of increasing severity, severity levels are **I** (informational), **W** (warning), **E** (recoverable error, and **S** (severe error). Another level, **M**, designates messages that are always written to the interface. (Level **M** does not imply an abnormal or error situation.) Console messages are always written to the trace file, regardless of message severity levels, if trace has been activated.
CONNECTEXIT	**METCNCTX**	Specifies the name of the exit routine to be executed when a user wants to connect to a service. This routine authenticates the user ID and password and can be used to set up information for customer billing and charge-back algorithms. The default routine (**METCNCTX**) provided with Host Services uses **RACROUTE** for user authentication.

Table 3 (Page 3 of 7). Parameters in the Control member

Keyword	Default	Description
DB2PLAN	**—**	Example: **DB2PLAN DSN METPLAN**. Specifies for a DB2 subsystem the prefix for Host Services application plan names. Parameter 1 to the DB2PLAN keyword is the DB2 subsystem name; parameter 2 is the plan name prefix. SQL routines append **C** or **R** to the plan-name prefix to generate the full plan name. Therefore, the plan names for the example are **METPLANC** and **METPLANR**. (For more information, see "SQL Service-Related Parameters" in Chapter 5 of *Installing Host Services and Configuring Host-to-LAN Communications for MVS*.)
DB2SS	**DSN**	Specifies the default subsystem name that Host Services is to use when a user connects to DB2 and either specifies the DB2 subsystem name **DBS** or does not explicitly specify a DB2 subsystem name.
DEGREE-ANY	**NO**	Specifies whether the SQL service should issue the SQL command **SET CURRENT DEGREE='ANY'** each time it opens a thread. There is some processing overhead associated with this parameter, but setting it can be beneficial for certain queries.
GATEWAY	**—**	Specifies the name of a gateway configuration member within the configuration data set. The configuration data set must have one such member for each gateway. Also the Control member must have one **GATEWAY** line for each gateway if communications with that gateway are to start automatically.
HOST-CODE-PAGE	**037**	Specifies the 3-digit number of the code page that Host Services is to use when converting character data. Any code-page specification in a gateway configuration member has precedence over the corresponding code-page specification in the Control member. The default for this field is 037, which specifies the EBCDIC code page for United States English, Canadian, Dutch, and Brazilian Portuguese.
INTERNALTRACE	**0**	Specifies the number of entries, in the internal trace table, to allocate per task. These entries are maintained as circular buffers. This feature is usually enabled only if you are experiencing difficulties with normal logging and tracing.

Table 3 (Page 4 of 7). Parameters in the Control member

Keyword	Default	Description
LOGLEVEL	**I**	Specifies the minimum severity level for messages to be written to the Host Services log. In order of increasing severity, severity levels are **I** (informational), **W** (warning), **E** (recoverable error, and **S** (severe error). Another level, **M**, designates messages that are always written to the log. (Level **M** does not imply an abnormal or error situation.) Log messages are always written to the trace file, regardless of message severity levels, if trace has been activated.
LOGMODE	**—**	Specifies the name of the VTAM logon-mode table entry.
LOGPOOL	**5 5**	Specifies the primary and secondary number of log and trace message blocks (LTMBs) for each allocation of a cell pool in extended common service area (ECSA) containing LTMBs. If allocation of a secondary cell pool would cause **MAXECSA** to be exceeded, the expansion is denied and the task waits for one second before retrying.
MAXECSA	**50**	Specifies the maximum amount of ECSA that Host Services can allocate, in units of 1024 bytes (1 KB). A value of 0 (zero) specifies no artificial restriction on ECSA allocation. The **MAXECSA** limit does not include ECSA allocations by other MVS facilities, such as VTAM and DB2. **Attention:** Be sure that your system has enough ECSA available to meet the specified amount.
MAXSERVICES	**0**	Specifies the maximum number of service address spaces that the service manager address space can start. If a gateway tries to start another service address space after the **MAXSERVICES** limit is reached, the request is denied. The default value of 0 (zero) sets the limit at the number of Host Services transaction program (TP) names that are defined in the gateway configuration members.
MESSAGE-CASE	**MIXED**	Must be either **MIXED** or **UPPER**, to specify whether interface and log messages are to be displayed in mixed-case characters or are to be translated to uppercase before being displayed.

Table 3 (Page 5 of 7). Parameters in the Control member

Keyword	Default	Description
OPERPOOL	**2 2**	Specifies the primary and secondary number of operator-command input blocks (OCIBs) for each allocation of a cell pool in ECSA containing OCIBs. If allocation of a secondary cell pool would cause **MAXECSA** to be exceeded, the expansion is denied and the operator is told to retry later.
PERM-CONNECT	**NO**	Specifies whether a SQL service address space's connection to DB2 should last for the lifetime of the address space. Specifying **YES** minimizes resource consumption when a user connects to DB2. The number of DB2 connections is limited, however; any DB2 **-DISPLAY THREAD(*)** command displays the idle permanent connections.
REQUIRE-DPF	**NO**	Specify **YES** or **NO**. The Desktop Password Facility (DPF) provides a secure method of ensuring that the user of a data access icon is authorized to use that icon. This option allows the IDS system administrator to require that all users connecting to this system use DPF.
ROUTCDE	**2**	Specifies which interfaces are to receive interface messages. The default (2) specifies Master Console Information. You can specify 1 to 15 interface numbers, separating them by commas or spaces as in this example: ROUTCDE 2,3 5
SQLEXIT	**METSQLX**	Specifies the name of the exit routine that is to be executed before and after each SQL statement received from the workstation. A value of **NONE** specifies that this routine is not required. For information about this exit, see the comments in the sample exit routine (**METSQLX**) and the discussion on specifying user exits in *Installing Host Services and Configuring Host-to-LAN Communications for MVS*.
SQSMEM	**100**	The number of kilobytes of dynamic storage to be used by SQL data encoding/decoding. Transferring complex data between the host and workstation might require more dynamic storage than provided by default.

Table 3 (Page 6 of 7). Parameters in the Control member

Keyword	Default	Description
SQSPROC	**METSQS**	Specifies the name of the cataloged procedure that the service manager task uses to start an SQL service address space. Use of different Host Services procedures lets the **PERFORM** parameter (on the **EXEC** statement) specify different performance groups for different Host Services subsystems or gateways. It also permits use of alternate **STEPLIB** s for testing subsystems or gateways. You can specify this cataloged procedure in either the Control member or a gateway configuration member.
SRBPOOL	**10 10**	Specifies the primary and secondary number of service request blocks (SRBs) for each allocation of a cell pool in ECSA containing SRBs. If allocation of a secondary cell pool would cause **MAXECSA** to be exceeded, the allocation is denied and the allocation request is either aborted with an error message or retried (depending on the function being executed).
STATUSEXIT	**NONE**	Specifies the name of the exit routine that is to report Host Services status information periodically. The user exit routine specified in **STATUSEXIT** formats and prints Host Services status information at time intervals specified in the **STATUSINTERVAL** parameter. A value of **NONE** specifies that this routine is not required. The routine returns 0 (zero) if it has handled all status information; or returns non-zero if Host Services should also write its usual status information. The routine must be written in assembler code. For other information, see the comments in the sample exit routine (**METSTATR**) and the discussion on specifying user exits in *Installing Host Services and Configuring Host-to-LAN Communications for MVS*.
STATUSINTERVAL	**900**	Specifies the number of seconds the system should wait between status reports. A user exit routine, specified in the **STATUSEXIT** parameter, formats and prints Host Services status information at time intervals specified in **STATUSINTERVAL**. The default value (**900**) specifies an interval of 15 minutes. Specifying **0** (zero) or **OFF** turns off the reporting facility.

Table 3 (Page 7 of 7). Parameters in the Control member

Keyword	Default	Description
SVC-CALLER-VERIFICATION	**OFF**	Must be **ON** or **OFF**. If you specify **ON**, the Host Services supervisor call (SVC) validates that the caller of the SVC is either a Host Services address space, the interface address space, the master scheduler address space, or the primary subsystem address space. With **ON** specified, TSO users cannot enter Host Services commands through the System Display and Search Facility (SDSF) or similar facilities. Failed validation causes the SVC to return an error code and to not perform the function.
SVCNUMBER	**224**	Specifies the SVC number for Host Services routines to use when performing authorized functions. The **METSVC** routine must be installed in **SYS1.LPALIB** as a type-3 SVC. The SVC number named in the subsystem specification (**IEFSSN xx**) is the one used to start the service manager address space; therefore, specifying the **SVCNUMBER** parameter in the configuration data set is useful only for testing a new SVC after the service manager address space has started.
TANPOOL	**8 8**	Specifies the primary and secondary number of task action/notification blocks (TANBs) for each allocation of a cell pool in ECSA containing TANBs. If allocation of a secondary cell pool would cause **MAXECSA** to be exceeded, the expansion is denied and the task needing to queue the action or notification can choose to retry or to continue processing.THREAD
	CONNECT	Must be **CONNECT** or **LUW**. If you specify **CONNECT**, the DB2 thread required for SQL requests is opened when the workstation user connects to DB2 and is closed when the user disconnects. With **LUW**, the DB2 thread is opened and closed at the beginning and end of each logical unit of work.
TRACE-CLASS	**A**	Specify the **SYSOUT** class that is to be used to allocate Host Services trace data sets when the **!TRACE START** command is used. Use **A-Z** or **1-9**. You cannot use *.

Gateway Communications Parameter Reference

Table 4 on page 57 lists the parameter keywords in a gateway configuration member, together with each parameter's description and legal values. None of these parameters has a default value.

To reconfigure values in a gateway configuration member, edit the appropriate member and then use **!STOP** and **!START** commands from the MVS interface to stop and restart communications with the corresponding gateway.

Table 4 (Page 1 of 2). Parameters in the gateway configuration member

Keyword	Description
CASPROC	Specifies the name of the cataloged procedure that the service manager task uses to start a CAS service address space. Use of different Host Services procedures lets the **PERFORM** parameter (on the **EXEC** statement) specify different performance groups for different Host Services subsystems or gateways. It also permits use of alternate **STEPLIB** s for testing subsystems or gateways. You can specify this cataloged procedure in either the Control member or a gateway configuration member. A **CASPROC** specification in a gateway configuration member has precedence over the **CASPROC** specification in the Control member.
CPREFIX	Specifies the 2-letter prefix for the Host Services tasks' TP names. This prefix is **SQ** for SQL (DB2) gateways or **CA** for CAS gateways. The format of Host Services TP names is described in “Sample Configuration Members” on page 49.
GATEWAY-ID	Specifies the 3-character database gateway identifier. The gateway identifier is used in forming host service and service manager TP names and in the syntax of the Host Services commands **!DISPLAY**, **!LOGON**, and **!STOP**. It must be exactly three characters in length, and must be unique for each gateway connecting to the same MVS system.
GATEWAY-TYPE	Specifies whether this gateway is a CAS gateway (**CAS**) or an SQL (DB2) gateway (**SQS**).
HOST-CODE-PAGE	Specifies the 3-digit number of the code page that is used for character data. Any code-page specification in a gateway configuration member has precedence over the corresponding code-page specification in the Control member.
PARTNER-LU-FOR-GTH	Specifies the remote LU name and the PU type for the APPC conversation through which the service manager task receives commands from the gateway. These commands, including **START** and **STOP**, are described in the IDS publication Configuring and Administering Database Gateway Services. The PU type must be 2.1.
PARTNER-LU-FOR-HTG	Specifies the remote LU name and the PU type for the APPC conversation through which the Monitor task sends “Service started” and“Service stopped” messages to the database gateway. The PU type must be 2.1.

Table 4 (Page 2 of 2). Parameters in the gateway configuration member

Keyword	Description
SERVICE LU(S)	Each line following this header specifies the application ID name, remote LU name, and PU type for a specific service address space. For example: METGWA 1 LUGWA PU2.1 / SERVICE 1 / METGWA 2 LUGWA PU2.1 / SERVICE 2 / The host service LU lines must be the last lines in a gateway configuration member. The PU type must be 2.1.
SQSPROC	Specifies the name of the cataloged procedure that the service manager task uses to start an SQL service address space. Use of different Host Services procedures lets the **PERFORM** parameter (on the **EXEC** statement) specify different performance groups for different Host Services subsystems or gateways. It also permits use of alternate **STEPLIB** s for testing subsystems or gateways. You can specify this cataloged procedure in either the Control member or a gateway configuration member. An **SQSPROC** specification in a gateway configuration member has precedence over the **SQSPROC** specification in the Control member.

Appendix C. Using the SQLMACRO Facility

With IDS, IBM introduces a new Host Services SQLMACRO facility. Many of the DIS Version 1.3.**x** gateway macros have been re-implemented using the SQLMACRO facility. As a result, operations such as creating a WTDD will execute much more quickly and reliably. While IDS continues to support gateway macros that send SQL to database systems for execution, the SQLMACRO facility provides significant advantages over the standard IDS gateway macro set.

The SQLMACRO facility permits conditional execution of SQL statements (**if**...**then**...**else**). It also permits checking of SQL return codes. Gateway macros do not provide this level of flexibility.

If generated by a gateway macro, each SQL statement and its result must be sent over the network. Gateway macros with many SQL statements, such as those required for building a WTDD, can be inefficient. The SQL macros controlled by the SQLMACRO facility reside in a data set on the MVS host system; the data set is accessible to the SQL services as well as batch.

While gateway macros must be executed from an IDS SA tool, the SQLMACRO facility permits execution from a variety of tools. The command SQLMACRO can be entered from any IDS tool that allows direct entry of SQL statements. In addition, the macros run by the SQLMACRO command can also be executed in batch mode on the MVS host system through the METWTBLD cataloged procedure. See **met**.JCL(WT*) for examples of executing the macros in batch mode.

The syntax of the SQLMACRO command issued from IDS is:

SQLMACRO **macro_name** [**parm_1** =**value_1**]...[**parm_n** =**value_n**]

where:

macro_name is the member name of a macro in the **met**.SQLMACRO data set.

parm_1 is the name of the first parameter.

value_1 is the value of the first parameter.

parm_n is the name of the **n** th parameter.

value_n is the value of the **n** th parameter.

Upon examining the member names in **met**.SQLMACRO you will find some macro names that begin with DSN, ARI, or QSQ. Macros with these names are run against DB2, SQL/DS, and OS/400 systems, respectively. The non-DB2 macros present in the SQLMACRO data set allow you to specify DRDA connections to non-DB2 systems and thereby execute SQL on those systems.

To see the intermediate results of SQLMACRO execution, examine the SYSOUT generated by SQL service address space. The SYSOUT class is the same as specified by the TRACE-CLASS configuration option. These intermediate results are present in the job output for batch jobs.

Appendix D. Handling Messages

Host Services generates status and error messages when necessary to alert you to problems or conditions that need attention. Some of these messages appear in only one place, a trace file. Others appear in both the trace file and the service manager log file. Others appear in both of these files and on the MVS host interface. The appearance of messages is controlled by the **CONLEVEL** and **LOGLEVEL** configuration parameters. Refer to Appendix B, "Configuration Parameter Reference" on page 49," and the IDS publication *Installing Host Services and Configuring Host-to-LAN Communications for MVS* for more information about these configuration parameters.

To interpret any status or error message, refer to the IDS *Messages and Codes*. In *Messages and Codes*, words in italic type are variables representing names or numbers reported within the messages. For example, in the following message, **string** is the command that is in error:

94.236 13.14. MCM2 57M ! METOPER **string** is not a valid Host Services command

Identifying Log and Trace Messages Generated by Host Services

The *Messages and Codes* lists messages generated by Host Services that can appear on the MVS interface and in the log file and trace file. The log file is governed by the **!WRITELOG** command, as described in the preceding chapter. The trace file is governed by the **!TRACE** command, also described in the preceding chapter. The **CONLEVEL** configuration parameter controls which operations-related messages appear on the MVS interface. The **LOGLEVEL** configuration parameter controls which messages appear in the Host Services log file.

Each message is preceded by the date and time it was generated. The date is expressed as year and day of year, and the time is expressed in hours, minutes, and seconds, as shown in the following example:

94.236 13.14. MCM2 59I ! METOPER IDS Host Services P command executed.

The sample message shown above was generated on the 236th day of 1994, at exactly 1:14 p.m.

The message itself begins with a message identifier comprising the Host Services component (**MCM**), four decimal digits (**2059** in the example), and a 1-character severity flag (**I** in the example).

The severity flag is **M**,**T, I**, **W**, **E**, or **S** :

M indicates an informational message that always appears on the interface, regardless of the **CONLEVEL** setting.

T indicates a trace message.

I indicates an informational message.

W identifies a warning message.

E identifies a recoverable error.

S indicates a severe error.

The exclamation mark (!) that follows the message number and severity flag in the preceding example indicates that the message was generated from the IDS Host Services subsystem whose subsystem command recognition character has been designated as "!".

Not all of the messages listed in *Messages and Codes* appear on the interface. The configuration file includes a **CONLEVEL** flag through which you indicate the lowest message severity to be displayed. For example, if you set the flag to **E**, the interface displays **S** and **E** messages but not **W** or**I** messages.

After the identifier, each message shows its origin. If the message came from the service manager address space, as in the previous example, it shows the name of the task that issued the message (**METOPER**, in this case). If the message came from a service address space, it shows the corresponding TP name (**SQGWA004**, for example).

In many cases, the most recent message is preceded by other messages that indicate the underlying error more specifically.

Contacting Support

In some cases, the problem solution requires that you contact your support organization. The solution will come more quickly if you follow these steps first:

1. If the error is reproducible and the log was not active: use the **WRITELOG** command to start the logging facility, and then reproduce the error.
2. Print a listing of the current log file and interface messages.
3. Show the log and interface listings to your support organization.

If your problem requires assistance beyond the capabilities of your own support group, consult the preface for detailed instructions on obtaining telephone assistance from IBM Technical Support.

Glossary

1. The ANSI/EIA Standard -- 440-A, **Fiber Optic Terminology**. Copies can be purchased from the Electronic Industries Association, 2001 Pennsylvania Avenue, N.W., Washington, DC 2006. Definitions are identified by the symbol (E) after the definition.
2. The definitions taken from draft international standards, committee drafts, and working papers being developed by OSO/IEC JTC1/SC1 are identified by the symbol (T) after the definition, indicating that the final agreement has not yet been reached among the participating National Bodies of SC1.

abend. (1)Abnormal end of task. (2)Synonym for abnormal termination.

abnormal end of task (abend). Termination of a task before its completion because of an error condition that cannot be resolved by recovery facilities while the task is executing.

abnormal end termination. (1)The cessation of processing prior to planned termination. (T)(2) A system failure or operator action that causes a job to end unsuccessfully. (3) Synonymous with abend, abnormal end.

access method. The way in which an application reads from or writes to a resource.

address space. (1) A range of virtual storage addresses created and maintained by the MVS operating system for data and program residence and for program execution; also, the program executing in an address space. MVS allocates real storage, processor, and input/output resources to an address space as needed to execute a program in that address space. MVS can maintain and control a large number of address spaces concurrently. (2) The range of addresses available to a computer program. (A)

Advanced Program-to-Program Communications (APPC).. An implementation of the SNA/SDLC LU 6.2 protocol that allows interconnected systems to communicate and share the processing of programs.

API. See application program interface.

APPC. See Advanced Program-to-Program Communications.

application. (1) A collection of software components used to perform specific types of user-oriented work on a computer. The terms application, application program, program, and tool are sometimes used interchangeably. See also tool.

application plan. In the MVS environment, the DB2 control structure produced during the bind process and used by the DB2 program to process the SQL statements encountered while an application is running.

application program interface (API). The interface through which an application program interacts with a service. An API usually consists of a set of functions or commands.

Application System/400. See AS/400.

AS/400. The IBM Application System/400 computer, which runs the OS/400 operating system.The AS/400 computer is classified as a midrange system.

asset. See asset management.

asset management. Identification and distribution of purchased IDS features, services, and tools licensed for use.

bind. In SQL, the process by which the output from the SQL precompiler is converted to a usable structure called an access plan.This process is the one during which access paths to the data are selected and some authorization checking is performed.See also application plan.

bit. Either of the digits 0 or 1 when used in the binary numeration system.Synonymous with binary digit.(T)

button. An object, in a window or window header, that contains graphics or text and is used to open other windows, open a menu, perform an action, or set a condition.

byte. A string that consists of a particular number of bits, usually eight, that is treated as a unit, and that represents a character.

CAF. See call attachment facility.

call attachment facility (CAF).. A DB2 attachment facility for application programs running in TSO or MVS batch. The CAF is an alternative to the DSN command processor and thus allows greater control over the execution environment.

Capsule. A tool for building an automated processing sequence using other icons.

CAS. See Cooperative Application Services.

CAS service address space. See service address space.

catalog. (1)A directory of files and libraries, with reference to their locations. A catalog can contain other information such as the types of devices in which the files are stored, passwords, and blocking factors. (I)(A)(2)To enter information about a file or a library into a catalog. (I)(A)(3)The collection of all data set indexes that are used by the control program to locate a volume containing a specific data set. (4)To include the volume identification of a data set in the catalog. (6)In SQL, tables, maintained by the database manager, that contain descriptions of objects, such as tables, views, and indexes.

cataloged data set. A data set that is represented in an index, or hierarchy of indexes, in the system catalog; the indexes provide the means for locating the data set.

cataloged procedure. A set of control statements placed in a library and retrievable by name.

class of service (COS). In SNA, a designation of the path control network characteristics, such as path security, transmission priority, and bandwidth, that apply to a particular session. The end user designates class of service at session initiation by using a symbolic name that is mapped into a list of virtual routes, any one of which can be selected for the session to provide the requested level of service.

client. A process, such as a program, that requests a service from another program.

CLIST. See command list.

code page. (1) An assignment of graphic characters and control function meanings to all code points; for example, assignment of characters and meanings to 256 code points for an 8-bit code, assignment of characters and meanings to 128 code points for a 7-bit code. (2) A particular assignment of hexadecimal identifiers to graphic characters.

command list (CLIST).. A list of MVS commands and statements designed to perform a specific function for the user under TSO.

command script. The software interface used for administering and managing a service.

common service area (CSA). In the MVS operating system, a part of the common area that contains data areas addressable by all address spaces.

Communications Manager/2. A licensed IBM program that runs under the OS/2 operating system and lets a workstation connect to a host computer and use the host resources as well as the resources of other personal computers to which the workstation is attached, either directly or through a host system.

interface. The software interface used for administering and managing a service.

conversation. In Systems Network Architecture (SNA)communications, a connection between two APPC transaction programs (TPs)over an LU-LU session that allows them to communicate with each other.

Cooperative Application Services (CAS). A combination of workstation software (a Cooperative Application Services transformer that runs within a capsule), a Cooperative Application Services gateway, and Host Services software running on an MVS or VM host. These three components allow IDS users to execute command scripts (such as those written in REXX)on the host and retrieve the results for further processing on their desktops.

COS. See class of service.

CSA. See common service area.

database administrator (DBA). Cooperative Application Services (CAS). The person who issues administrative instructions to a database service. The DBA defines the data tables used by data access tools, monitors database performance, and assigns user access.

DATABASE 2. See DB2.

Data Entry. The tool that is used to enter, retrieve, delete, and update data on a database server, or to set @-variables in the User Input Controls window of a capsule application.

data set. In the MVS environment, the major unit of data storage and retrieval, consisting of a collection of data in one of several prescribed arrangements and described by control information to which the system has access.

DBA. (1)See database administrator. (2)A tool that allows a database administrator to define data types, structure, and arrangement in a logical database or

Workstation Tools Data Dictionary (WTDD)and, from the logical definitions, create physical tables that will contain the actual data. See also Workstation Tools Data Dictionary.

DB2. DATABASE 2, an IBM relational database management system that runs on an MVS system.

DD. In JCL, a data definition statement that identifies and describes a data set, and specifies the input and output resources needed for the data set.

desktop. The screen space occupied by IDS. See also OS/2 desktop.

IDS. The IBM Intelligent Decision Server.

disable. (1)To make nonfunctional. (2)In interactive communications, to disconnect or stop a subsystem. Contrast with enable.

distributed facility. A software component composed of parts that physically reside and function on more than one computer in a computer network.

Distributed Relational Database Architecture (DRDA). An IBM architecture that allows SQL data requests between DRDA data sources within a standard protocol. The database becomes a coordinator to SQL requests to other DRDA data sources

DRDA. See Distributed Relational Database Architecture.

dump. (1)To record, at a particular instant, the contents of all or part of one storage device in another storage device. Dumping is usually for the purposes of debugging. (T)(2)Data that has been dumped. (T)(3)To copy data in a readable format from main or auxiliary storage onto an external medium such as tape, diskette, or printer. (4)To copy the contents of all or part of virtual storage for the purpose of collecting error information.

dynamic storage. (1) A device that stores data in a manner that permits the data to move or vary with time such that the specified data are not always available for recovery. Magnetic drum and disk storage are dynamic, nonvolatile storage. An acoustic delay line is a dynamic, volatile storage. (A) (2) A storage in which the cells require repetitive application of control signals to retain stored data. Such repetitive application of the control signals is normally called a refresh operation. A dynamic storage can use static addressing or sensing circuits. (A) See also dynamic storage allocation.

dynamic storage allocation. A storage allocation technique in which the storage areas assigned to computer programs and data are determined by criteria applied at the moment of need. (I) (A)

EBCDIC. Extended binary-coded decimal interchange code. A coded character set consisting of 8-bit coded characters.

ECSA. See extended common service area.

ELPA. See Extended Link Pack Area.

enable. (1)To make functional. (2)In interactive communications, to load and start a subsystem. Contrast with disable.

Ethernet. A type of local area network used for IDS connections. An Ethernet network uses a bus topology (that is, all devices are connected to a single cable). An Ethernet network can include any of several kinds of physical transmission media.

extended architecture. An extension to System/370 architecture that takes advantage of continuing high performance enhancements to computer system hardware.

extended common service area (ECSA). In the MVS operating system, an extension to the common service area, which resides above the 16-Mbyte line. See also common service area.

Extended Link Pack Area (ELPA). In the MVS operating system, an extension to the Link Pack Area, which resides above the 16-Mbyte line. See also Link Pack Area.

file. A collection of information stored as a unit on a disk or other storage media. See also data set, executable file, directory.

gateway. A software process that translates protocols and access methods between two dissimilar systems. See also access method.

gateway identifier. A unique 3-character string, such as GWA or GWB, that identifies a gateway service as used in various configuration parameters.

gateway name. The name by which a gateway is known to other components on the local area network (LAN).

gateway service. A IDS facility comprised of one or more gateways of a particular type. See also gateway.

GRANT statement. A DB2, process that authorizes users to access data.

host service. A IDS Host Services address space on the MVS host that controls access to DB2 on behalf of database gateways and users or Cooperative Application Services.

Host Services. The IDS program product residing on a host computer. Host Services provides interface services between IDS on the LAN and the database or CAS application on the host computer.

input. (1)Data that is entered into a computer for processing. (2)Data received from other tools, such as Spreadsheet or Query. (3) Contrast with output. See also input region.

input/output device. Synonym for input/output unit.

input/output unit. A device in a data processing system by means of which data can be entered into the system, received from the system, or both. (I)(A)Synonymous with input/output device.

Interactive System Productivity Facility (ISPF). An IBM licensed program that serves as a full-screen editor and dialogue manager. Used for writing application programs, it provides a means of generating standard screen panels and interactive dialogues between the application programmer and terminal user.

internetwork. Two or more local area networks (LANs)interconnected for exchange of data.

ISPF. See Interactive System Productivity Facility.

JCL. Abbreviation for job control language, a control language used in an IBM mainframe environment to identify a job to an operating system and to describe the job's requirements.

job control language. See JCL.

KB. See kilobyte; 1024 bytes. See also byte.

kilobyte (KB). 1024 bytes. See also byte.

LAN. See local area network.

local area network (LAN).. A data communications network, spanning a limited geographical area, in which serial transmission is used for direct communication among computers or other devices. See also serial transmission.

local network. The network to which a specified user or resource is directly connected. See also remote network.

log. A collection of messages or other information that is stored for data collection or troubleshooting purposes.

logical unit (LU). In Systems Network Architecture (SNA)communications, a port through which a user or program accesses the SNA network for communication with another user or program. Contrast with physical unit.

logical unit 6. 2 (LU 6.2.). A set of protocols and services defined by IBM's SNA for communication between application programs. See also Advanced Program-to-Program Communications (APPC).

LU. See logical unit.

LU-LU session. In Systems Network Architecture (SNA)communications, a session between two logical units (LUs)in an SNA network. It provides communication between two end users, or between an end user and an LU services component.

LU 6. 2. See logical unit 6.2.

member. A file within a partitioned data set.

Multiple Virtual Storage (MVS).. A facility that allows each user a private address space. Multiple Virtual Storage consists of MVS/370 Data Facility Product and the MVS/370 Data Facility Product operating on a System/370 or ES/9000 processor.

Multiple Virtual Storage/Enterprise Systems Architecture. See MVS/ESA.

MVS. Multiple virtual storage. Implies the MVS/ESA product.

MVS/ESA. Multiple Virtual Storage/Enterprise Systems Architecture. An IBM MVS operating system environment that supports IBM S/390 and ES/9000. MVS/ESA allows an address space access to one or more data spaces. A data space is, like an address space, a virtual storage area. While an address space contains system programs, user programs, and data, a data space can contain only data. Each data space can provide up to two gigabytes (two billion bytes)of virtual storage for data. Data spaces can save time, prevent unintentional changes to data, and permit the sharing of data among selected users. So, in addition to the two gigabytes a job has available to it for holding instructions and data in an

address space, each job can also have access to many data spaces.

NAU. See network addressable unit.

network. A set of workstations, servers, and other devices connected together for the interchange of information. See also local area network.

network addressable unit (NAU).. In SNA, a logical unit. The NAU is the origin or the destination of information transmitted by the path control network.

node. In communications, an end point of a communications link or a junction common to two or more links in a network. Nodes can be processors, workstations, or other devices. Nodes can vary in routing and other functional capabilities.

OCIB. operator-command input block.

Operating System/400. See OS/400.

Operating System/2. See OS/2.

operator-command input block (OCIB). An internal control block used in conjunction with processing MVS operator commands directed to Host Services code.

OS/400. Operating System/400. The operating system for the IBM AS/400 computer.

OS/2. Operating System/2. A single-user, mulitasking, IBM operating system that runs on microcomputers with an Intel 80286 or later microprocessor.

output. The result of computer processing.

pacing. See session-level pacing.

parallel sessions. In SNA, two or more concurrently active sessions between the same two logical units (LUs)using different pairs of network addresses. Each session can have independent session parameters.

parameter. Data passed by a user to a program to direct or limit processing in some way.

partitioned data set. In MVS, a data set, in direct-access storage, that is divided into files called members, each of which can contain a program, part of a program, or data. See also data set, member.

partner logical unit (PLU). In Systems Network Architecture (SNA)communications, the remote, corresponding participant in a session in an SNA network.

password. A unique string of characters known to a computer system and to a user, who must specify the character string to gain access to a desktop, gateway service, database, system, or service within a system. See also user ID.

path. A single logical connection between two points, such as between a Host Service and a LAN service.

PC. See personal computer.

personal computer (PC). A microcomputer designed primarily for use by a single user.

physical unit (PU). In Systems Network Architecture (SNA), a network-addressable unit that manages and monitors the resources of a node. Contrast with logical unit.

plan. See application plan.

PLU. See partner logical unit.

pool. A collection of similar gateways that access a single RDBMS or host application to equalize the loads of the pool's member services.

privilege. In SQL, a capability given to a user by the processing of a GRANT statement.

PU. See physical unit.

query. A request for data from a database, based on specified conditions.

Query. A tool used to retrieve data from a database, based on specified conditions.

RDBMS. See relational database management system.

realm. In IDS networks, a collection of IDS resources that are maintained as a logical group (that is, a group that is determined not by physical boundaries, but rather by virtue of being associated with one another). A realm must contain at least a Primary File Service and a Directory Service. Other components such as workstations, Internetwork Routing Service, and Data Access Services can be added to form a functional IDS local area network (LAN).

relational data. In a relational database, information that is related to other information and whose relationships can be defined in the form of tables consisting of rows and columns of data.

relational database. A database that is organized and accessed according to relationships between data items. These relationships are expressed as tables related by data values, rather than by pointers or memory locations.

relational database management system (RDBMS).. The software that makes it possible to create, use, and maintain relational data in the form of tables consisting of rows and columns of data.

remote network. A network other than the local network, but which is accessible to the local network by some means. See also local network.

REXX. IBM's Restructured Extended Executor language, a command language in which scripts can be written for use in Cooperative Application Services and Global Access for VM.

script. See command script.

SDLC. See Synchronous Data Link Control.

SDSF. See System Display and Search Facility.

serial transmission. The sequential transmission of one data bit at a time.

server. The computer on which a service runs.

service. A software process that provides facilities to other parts of IDS, such as a file service, a mail service, a mail gateway service, and a data access service. In IDS Host Services for MVS, each host service runs in a service address space.

Service Global Data Area (SGDA). An Extended Common Service Area (ECSA)control block used by IDS Host Services to provide global information about a host service. This includes information such as the TP name for the APPC conversation, the gateway to which the service belongs, and end user connection status. There is one SGDA per host service.

service manager. An MVS Host Services address space that coordinates one or more sets of subsystem services.

service request block (SRB). An MVS control block used to communicate work requests between different address spaces.

session. A logical connection between two network addressable units (NAUs)that allows communication between the NAUs.

session level-pacing. In SNA, a flow control technique that permits a receiving session to control the data transfer rate (the rate at which it receives request units)on the normal flow. It is used to prevent overloading a receiver with unprocessed requests when the sender can generate requests faster than the receiver can process them.

session limit. (1)In SNA, the maximum number of concurrently active LU-LU sessions that a particular logical unit (LU)can support. (2)In the network control program, the maximum number of concurrent line-scheduling sessions on a non-SDLC, multipoint line.

session mode name. In SNA, defines certain session characteristics like pacing level and class of service.

session pacing. See session-level pacing.

SGDA. See Service Global Data Area.

Show Controls. A window header button, usually in the main window of an icon, that opens a controls window in which the user can specify settings that determine icon behavior, how a program is processed, or both.

SNA. See Systems Network Architecture.

SQL. Structured Query Language, a command language for use with relational databases. The language consists of statements to insert, update, delete, query, and protect data. IDS tools use SQL to communicate with database servers.

SQL/DS. The IBM Structured Query Language/Data System relational database management system that runs on a VM system.

SQL Entry. A tool used to run Structured Query Language (SQL)programs.

SRB. See service request block.

SSDA. See subsystem data area.

Structured Query Language. See SQL.

Structured Query Language/Data System. See SQL/DS.

subsystem. A secondary or subordinate system, usually capable of operating independently of, or asynchronously with, a controlling system.

subsystem data area (SSDA). A Host Services for MVS control block that describes each IDS subsystem.

supervisor call (SVC). A request that serves as the interface into operating system functions, such as allocating storage. The SVC protects the operating system from inappropriate user entry. All operating system requests must be handled by SVCs.

SVC. See supervisor call.

Synchronous Data Link Control (SDLC). A discipline conforming to subsets of the Advanced Data Communication Control Procedures (ADCCP)of the American National Standards Institute (ANSI)and High-level Data Link Control (HDLC)of the International Organization for Standardization, for managing synchronous, code-transparent, serial-by-bit information transfer over a link connection. Transmission exchanges can be duplex or half-duplex over switched or nonswitched links. The configuration of the link connection can be point-to-point, multipoint, or loop. (I)

SYSOUT. A system output stream, also an indicator used in data definition statements to signify that a data set is to be written on a system output unit.

SYSOUT class. A category of output with specific characteristics and written on a specific output device.

system administrator. In IDS, the person who is responsible for maintaining and configuring IDS, including its workstations, servers, and network.

System Display and Search Facility (SDSF). In the MVS and VM environments, a facility that provides current information on the status of jobs, print queues, system resources, and the like.

Systems Network Architecture (SNA).. The IBM description of logical structure, formats, protocols, and operational sequences for network configuration, operation, and transmission.

table. In a database, a single file, consisting of data organized in columns and rows.

TANB. See task/action notification block.

task/action notification block (TANB). In the MVS environment, internal control block used to communicate work requests between Host Services tasks.

Text. A tool used to write and format text documents, format the results of data access tools, and receive data and images from other tools and PC applications.

Time-Sharing Option (TSO). An MVS program product that provides interactive time sharing from local and remote terminals.

Time-Sharing Option Extensions(TSO/E). (1)The base for all TSO enhancements. It provides MVS users with additional functions, improved usability, and better performance. (2)In the MVS/ESA environment, TSO/E also provides virtual storage constraint relief.

token. In a local area network, the symbol of authority passed successively from one data station to another to indicate the station temporarily in control of the transmission medium. Each data station has an opportunity to acquire and use the token to control the medium. A token is a particular message or bit pattern that signifies permission to transmit. (T)

token-ring network. A network that uses a ring topology, in which tokens are passed in a circuit from node to node. A node that is ready to send can capture the token and insert data for transmission. See also token, node.

tool. A IDS workstation program, such as Text, Reporter, or Query. Every tool is represented by an icon.

TP. See transaction program.

transaction program (TP). A program that uses an APPC application programming interface to communicate with a partner program on a remote node.

transformer. A tool that can be used to perform various types of data manipulation either interactively or in a capsule application.

TSO. See Time-Sharing Option.

TSO/E. See Time-Sharing Option Extensions.

virtual machine (VM). (1)A virtual data processing system that appears to be at the exclusive disposal of a particular user, but whose functions are accomplished by sharing the resources of a real processing system. (T)(2) A system in which each user appears to have his own computer and input/output devices. (3)A functional simulation of a computer and its associated devices. Each virtual machine is controlled by a suitable operating system; for example, the conversational monitor system. VM/370 controls concurrent execution of multiple virtual machines on a single System/370 computing system. (4)In VM, a functional equivalent of either a System/370 computing system or a System/370-Extended Architec-

ture computing system. Each virtual machine is controlled by an operating system. VM controls concurrent execution of multiple virtual machines on a single system.

Virtual Telecommunications Access Method (VTAM).. A set of programs that maintain control of the communication between terminals and application programs running under DOS/VS, OS/VS1 and OS/VS2 operating systems.

VM. See virtual machine.

VTAM. See Virtual Telecommunications Access Method.

worksheet. A IDS Host Services installation and configuration planning tool. You can find planning worksheets in the IDS publication Planning for MVS Host-to-LAN Connectivity.

workstation. A computer at which the user can use IDS to perform work. Depending upon the model, workstation components include the monitor, keyboard, mouse, and system unit, which can house a disk drive.

workstation keyboard. The physical keyboard, which is the primary input device used to interact with IDS.

Workstation Tools Data Dictionary (WTDD).. A logical data dictionary that standardizes the IDS workstation view of data across many relational database management systems. The WTDD stores the print names of database items, defines how the items are to be displayed, and includes other logical attributes of data. The WTDD is implemented as a set of standard relational tables that can be managed through the DBA tool. See also print name.

WTDD. See Workstation Tools Data Dictionary.

IDS Publications

This section lists IDS publications, including order numbers. To order copies of the books listed here, or to get more information about a book, see your IBM representative.

General Publications

Getting Started with the Intelligent Decision Server (SC26-8892)

Working with the Intelligent Decision Server (SC26-8893)

Installation, Configuration, and Administration Publications

Installing IDS LAN Components (GC26-8877)

System Administration Guide and Reference (SC26-8890)

Messages and Codes (GC26-8878)

Database Gateway Services Publications

Installing Database Gateway Services for UNIX Environments (GC26-8884)

Configuring and Administering Database Gateway Services (GC26-8885)

Administering Databases with IDS (SC26-8891)

Host Services Publications

Planning MVS Host-to-LAN Connectivity (GC26-8880)

Installing Host Services and Configuring Host-to-LAN Communications for MVS (GC26-8881)

Administering Host Services for MVS: DB2 and Cooperative Application Services (GC26-8882)

Developing Applications with Cooperative Application Services (GC26-8883)

Configuring and Administering Mail Gateway Services (GC26-8923)

Open Data Access Service Publications

Developing Applications with Open Data Access Service (SC26-8888)

Data Access Tools Publications

Browser User's Guide (SC26-8900)

Data Entry User's Guide (SC26-8897)

DBA Tool User's Guide (SC26-8894)

SQL Entry User's Guide (SC26-8898)

Data Analysis Tools Publications

BASIC Tool User's Guide (SC26-8908)

Query User's Guide (SC26-8901)

Spreadsheet User's Guide (SC26-8904)

Presentation Tools Publications

Layout User's Guide (SC26-8902)

Plot User's Guide (SC26-8905)

Reporter User's Guide (SC26-8903)

Text User's Guide (SC26-8896)

Application Development Tools Publications

Transformers Guide (SC26-8907)
Capsule User's Guide (SC26-8906)
Forms Design User's Guide (SC26-8899)
Interacting with PC Applications (SC26-8895)
Installing and Configuring IDS Open Clients (GC26-8889)

Index

Special Characters

-DISPLAY THREAD(*) command 50
! command recognition character 24
!DISABLE ALL command 26, 28
!DISABLE command 25
!DISABLE gateway_identifier command 26
!DISABLE gateway_name command 26
!DISABLE service_number command 26, 28
!DISPLAY CAS command 27
!DISPLAY command 27, 57
!DISPLAY CONNECTED command 27
!DISPLAY gateway_identifier command 28
!DISPLAY gateway_name command 28
!DISPLAY GATEWAYS command 28
!DISPLAY OPTIONS command 28
!DISPLAY PLANS command 28
!DISPLAY service_name command 28
!DISPLAY SERVICES command 28
!DISPLAY SQS command 29
!ENABLE ALL command 31
!ENABLE command 30
!ENABLE gateway_identifier command 30
!ENABLE gateway_name command 30
!ENABLE service_number command 31
!LOGON command 57
!SET command 32, 49
!SET configuration_option command 32
!SET value command 32
!SNAP ALL command 33
!SNAP CAS command 33
!SNAP command 32
!SNAP DATASET dataset_name command 33
!SNAP ECSA command 33
!SNAP LOG command 33
!SNAP MANAGER command 33
!SNAP MASTER command 33
!SNAP MONITOR command 33
!SNAP OPERATOR command 33
!SNAP SERVICE ALL command 33
!SNAP SERVICE service_name command 34
!SNAP SGDA ALL command 34
!SNAP SGDA service_name command 34
!SNAP SQS command 34
!SNAP SSDA command 34
!SNAP STOP command 34
!SNAP SYSOUT class command 34
!START ALL command 38
!START COLD command 35
!START command 49, 57
!START control_member_name command 35
!START gateway_identifier command 37
!START gateway_name command 36, 37
!START METSERV command 35
!START service_number command 38
!STOP ALL command 39, 41
!STOP ALL NOPROMPT command 39
!STOP command 39, 49, 57
!STOP gateway_identifier command 39, 40
!STOP gateway_name command 39, 40
!STOP METSERV command 39
!STOP METSERV NOPROMPT command 39
!STOP service_name command 39
!STOP service_number command 41
!TRACE ALL command 42
!TRACE CAS command 42
!TRACE command 41, 45, 61
!TRACE gateway_identifier command 42
!TRACE gateway_name command 42
!TRACE LOG command 42
!TRACE MANAGER command 42
!TRACE MASTER command 42
!TRACE MONITOR command 42
!TRACE OPERATOR command 42
!TRACE service_name command 42
!TRACE SERVICES command 42
!TRACE SQS command 42
!TRACE START command 42, 56
!TRACE STOP command 43
!WRITELOG command 43, 45, 61
!WRITELOG dataset_name command 43
!WRITELOG STOP command 43
!WRITELOG SYSOUT class command 43
“Service started” message, issued by Monitor task 57
“Service stopped” message, issued by Monitor task 57

A

access, granting to DB2 data 13
active service address spaces, terminating 35
address space identifier (ASID), displaying for every service address space 28

address space, defined 9
ALL operand
used with !ENABLE 31
used with !SNAP 33
used with !START 38
used with !STOP 39, 41
used with !TRACE 42
APPC 2
communications between host and LAN 7
enabling conversations with the !START command 36
paths between Host Services and database gateway service 11
protocol provided by LU 6.2 8
application ID name
specifying for the Monitor task's APPC conversations 50
specifying for the service manager task's APPC conversations 50
application plans 50
creating 14, 15
displaying for DB2 configurations 28
specifying Host Services plan name prefix for a DB2 subsystem 50
APPLID
names 50
specifying for the Monitor task's APPC conversations 50
specifying for the service manager task's APPC conversations 50
APPLID-FOR-GTH parameter 50
APPLID-FOR-HTG parameter 50
ASID, see address space identifier 28
asset management process 4
ATTACH subtask, specifying timeout on completion of 50
ATTACHINTERVAL parameter 50
authenticating user ID and password 50
authorization IDs
configuring for DB2 13
displaying for a service address space 28

B

BIND job 15
BIND ON PLAN, for an existing plan 14
BINDADD, minimum privilege required for a new plan 14
binding, the DB2 application plans used by Host Services 13
buffers, circular 50

C

C 2
CA prefix 50
CAF 6, 10
cancel requests, managed by gateway monitor process 4
capturing data with the !SNAP command 32
CAS
accessing applications from IDS workstations 1
data access components 6
defined 1
service, function defined 6
CAS address spaces
capturing the global and local data areas for all 33
CAS gateway, configuration member 49
CAS operand
used with !DISPLAY 27
used with !SNAP 33
used with !TRACE 42
CAS service address spaces
activating tracing in all 42
function 10
introduced 2, 9
CAS sessions, controlling 3
CAS transformer
concurrent openings limits for 10
range of host software used with 2
case, specifying for interface and log messages 50
CASMEM parameter 50
CASPROC parameter 50, 57
cataloged data set, specifying for
capturing “snapshot” data 33
containing logged information 43
cataloged procedure, specifying for starting
CAS service address space 50, 57
SQL service address space 50, 57
Category field 20
cell pool 50
change configuration setting 32
character data, converting 50, 57
charge-back algorithms 50
circular buffers 50
class of service 8
class variable, used with !WRITELOG 43
CLIST
commands executed through TSO/E 6
executing in the same CAS service address space as TSO/E 10

CLIST *(continued)*
procedure sent by workstation 2
procedures invoking cooperative application programs 7
code page
specifying for Host Services 50
specifying in the gateway configuration member 57
COLD operand, used with !START 35
command recognition character 24
command reference 25
command syntax conventions 24
commands
automatically performing when the service manager starts 50
database gateway service 3
descriptions 25, 44
host interface 3, 23
in Control configuration member 50
including in the service manager configuration member 24
issuing from the host interface 23
summary list 23
syntax conventions 24
communications
between data access components 4
between data access tools and host 4
illustration
access path from IDS workstations to host 2
CAS communications system view 6
data access communications system view 4
service address space 9, 10
service manager address space for data access communications 11
interface 5
starting between a DB2 gateway and multiple service address spaces 36
stopping between a service manager and gateway 39
Communications Manager/2
host connections 2
participating as an SNA network node 7
role in IDS 4
configuration
APPLID-FOR-GTH parameter 50
APPLID-FOR-HTG parameter 50
ATTACHINTERVAL parameter 50
CASMEM parameter 50
CASPROC parameter 50, 57
CONLEVEL parameter 50
CONNECTEXIT parameter 50
configuration *(continued)*
CPREFIX parameter 57
DB2PLAN parameter 50
DB2SS parameter 50
default values for all configuration parameters 49
DEGREE-ANY parameter 50
GATEWAY parameter 50
GATEWAY-TYPE parameter 57
HOST-CODE-PAGE parameter 50, 57
INTERNALTRACE parameter 50
LOGLEVEL parameter 50
LOGMODE parameter 50
LOGPOOL parameter 50
MAXECSA parameter 50
MAXSERVICES parameter 50
MESSAGE-CASE parameter 50
OPERPOOL parameter 50
parameter reference 49
PARTNER-LU-FOR-GTH parameter 57
PARTNER-LU-FOR-HTG parameter 57
PERM-CONNECT parameter 50
REQUIRE-DPF parameter 50
ROUTCDE parameter 50
sample data set 49
sample members 49
SERVICE LU(S) parameter 57
SQL gateway configuration member 49
SQLEXIT parameter 50
SQSMEM parameter 50
SQSPROC parameter 50, 57
SRBPOOL parameter 50
STATUS-EXIT parameter 50
STATUSINTERVAL parameter 50
SVC-CALLER-VERIFICATION parameter 50
SVCNUMBER parameter 56
TANPOOL parameter 56
THREAD parameter 56
TRACE-CLASS parameter 56
configuration setting, changing with the !SET command 32
configuration_option variable, used with !SET 32
configuring, data access tools to use DRDA 47
CONLEVEL
configuration parameter 45, 61
flag 46, 62
CONLEVEL parameter 50
CONNECT, determining opening and closing of DB2 thread 56
connected authorization ID, displaying for a service address space 28

CONNECTED operand, used with !DISPLAY 27
CONNECTEXIT parameter 50
Control member
  configuration parameters in 50
  containing configuration information 49
  holding default values for all configuration parameters 49
  parameter reference 50
control_member_name variable, used with !START 35
controlling sessions 3
conversation
  defined 8
  one session supporting many conversations 8
converting data and format 4
CPREFIX
  parameter 57
  value, specified in the gateway configuration member 50
CREATEDBA
  minimum privilege required 16
  privilege 15
CREATESG, minimum privilege required 16
createsyn macro 21
createsynall macro 21
createsyndbmscat macro 21
createsynexplain macro 21
createsynwtdd macro 21
CSA 33
customer billing algorithms 50

D

data access components, examining communications between 4
Data Access Services, defined 1
data access tools
  using the WTDD to translate commands and data 6
data areas
  capturing
    for a service address space 34
    in the ECSA 33
    with the !SNAP command 32
  in ECSA, freeing all global 39
data conversion 4
data sets
  closing and releasing with the !SNAP STOP command 34
  required for data access preparation 16
database administrator
  creating for IDS 15
database administrator *(continued)*
  creating user ID for 16
  granting DBADM privileges on the database 16
database gateway service
  converting protocols through 7
  interface interface 3
  processing commands issued from the host interface 23
  role in data access 2
  using Communications Manager/2 to establish SNA sessions with 4
Database Logon field 20
Database Name field 20
database, steps for creating an IDS 15, 17
DATASET dataset_name operands, used with !SNAP 33
DATASET operand, used with !SNAP 33
dataset_name variable
  used with !SNAP 33
  used with !WRITELOG 43
DB2
  -DISPLAY THREAD(*) command 50
  access, granting to users 13
  accessing from IDS workstations 1
  address spaces 9
  authorization IDs, configuring 13
  Call Attach Facility (CAF) 6
  catalog tables, creating synonyms for 21
  data access components 5
  EXPLAIN table
    allowing user access to 21
    creating synonyms for 21
  EXPLAIN, loading a description in the WTDD 16
  gateway
    displaying 28
    stopping conversations for 39, 40
  sample table, loading a description into the WTDD 16
  sessions, controlling 3
  specifying the default subsystem name 50
  SYSTEM tables 21
    creating synonyms for 21
  thread, determining opening and closing of 56
DB2PLAN parameter 50
DB2SS parameter 20, 50
DBADM
  minimum privileges required 16
  privileges, granting on the database to the IDS database administrator 16

DBS 50
DEGREE-ANY parameter 50
Desktop Password Facility, see DPF 50
dis.JCL 16
dis.JCL(WTBLDUTL), editing 16
dis.JCL(WTCRTDB), editing 16
dis.JCL(WTLOAD), editing 16
dis.JCL(WTREORG), editing 17
disabling service address spaces 25
display interface information 27
DPF, requiring that all connecting users use 50
DRDA, accessing non-DB2 databases through 3, 47
DSN keyword 33, 50
dynamic storage, specifying for
  CAS data encoding/decoding 50
  SQL data encoding/decoding 50

E

E severity flag 46, 50, 62
EBCDIC code page 50
ECSA
  capturing the Host Services data areas in 33
  specifying allocation limits 50
  specifying number of LTMBs for each allocation of a cell pool in 50
ECSA operand, used with !SNAP 33
enabling service address spaces 30
encoding/decoding
  dynamic storage used by CAS 50
  dynamic storage used by SQL 50
Enterprise System/9000 1
error messages
  indicators 45, 50, 61
  interpreting 45, 61
errors, logging and reproducing 46, 62
ES/9000 1
Ethernet LAN 5
exit routine
  specifying for execution when user connects to a service 50
  specifying the name to be executed for SQL statements 50
EXPLAIN table
  allowing user access to 21
  creating synonyms for 21

F

fields
  Category 20
  Database Logon 20
  Database Name 20
  Gateway 20
  Password 20
  Secured 20
  SQL ID 20
  User Name 20
  WTTD Owner 20
format conversion 4

G

gateway
  activating tracing in all service address spaces for 42
  communications parameter reference 56
  configuration member parameter keywords list 56
  configuration members 49
  displaying the 3-character ID for each service address 28
  identifying
    to enable service address spaces 30
    to start a service manager 36
  maintaining APPC conversations to the host 4
  managing
    allocation of internal processes through the monitor process 4
    cancel requests for 4
  one line in the Control member for each gateway 50
  preventing from automatically restarting 41
  relationship with service address spaces 9
  specifying type in the gateway configuration member 57
  starting
    a service address space 50
    all service address spaces for 38
    communications automatically 50
    communications with 35
  statement in the service manager configuration member 37
  statements to specify gateway configuration members 49
  stopping
    conversations for a 39
    service address spaces for 40
  type, displaying for every gateway 28

gateway configuration
CAS member 49
SQL member 49
gateway configuration member
name 49
reconfiguring values in 49
specifying within the configuration data set 50
Gateway field 20
gateway identifier
default identifier 50
displaying
for a service address space 28
for every gateway 28
forming part of the TP name 50
specifying for starting service address spaces 37
specifying in the gateway configuration member 57
GATEWAY operand, used with !DISPLAY 28
GATEWAY parameter 50
gateway service
communications path to a service address space 9, 10
relationship with service address spaces 11
role in data access 2
gateway type, displaying for every gateway 28
GATEWAY-ID parameter 57
gateway-to-host communications path 11
GATEWAY-TYPE parameter 57
gateway_identifier variable
used with !DISABLE 26
used with !DISPLAY 28
used with !ENABLE 30
used with !START 37
used with !STOP 39, 40
used with !TRACE 42
gateway_name variable
used with !DISABLE 26
used with !DISPLAY 28
used with !ENABLE 30
used with !START 36, 37
used with !STOP 39, 40
used with !TRACE 42
global data areas
capturing for all service address spaces 33
in ECSA, freeing all 39
granting
access to DB2 13
all table privileges 14
DBADM privileges on the database to the IDS database administrator 16
GWA, as default database gateway identifier 50
GWASQS 49
GWB, as default CAS gateway identifier 50
GWBCAS 49

H

hexadecimal address space identifier, displaying for a service address 28
host interface
commands 23
controlling logging of messages 43
interface 3
interpreting messages 45, 61
Host Services
address spaces 9, 11
capturing SSDA with the !SNAP command 34
command summary list 23
configuration parameter reference 49
defined 1
identifiers 34
messages written to log 50
service address identifier number 50
subsystem 11
subsystem messages generated by 46, 62
host services, residing in MVS address spaces 6
HOST-CODE-PAGE parameter 50, 57
host-to-gateway communications path 11

I

I severity flag 46, 50, 62
IBM Technical Support 46, 62
Icon Options window 18, 20
ID
gateway 26, 28, 30, 37
service name 28
IDS OS/2 Database Gateway Service 2
informational message 46
informational message severity level 50, 62
installation
database gateway services 4
Host Services for MVS 1
planning for DB2 data access in an MVS environment 1
planning for SNA APPC communications in an MVS environment 8
interface
database gateway service interface 3
host interface 3

interface *(continued)*
interface 11
interpreting messages 45, 61
message indicator 50
messages
controlling logging of 43
printing a listing of 62
specifying case for 50
specifying which interfaces receive messages 50
interface messages
printing a listing of 46
internal trace table, specifying the number of entries to allocate per task 50
INTERNALTRACE parameter 50
interval status report 50

L

limiting sessions between LUs 8
loading, initial table descriptions into the WTDD 16
local data areas, capturing for all service address spaces 33
log and trace message blocks (LTMBs), specifying the primary and secondary number of 50
log file, printing a listing of 46, 62
log interface messages 43
log message indicator 50
log messages
specifying case for 50
written to trace file 50
LOG operand
used with !SNAP 33
used with !TRACE 42
Log task
activating tracing in the 42
capturing the data areas for 33
logging
experiencing difficulties with 50
messages, carried by SRBs across address spaces 11
specifying
the cataloged data set to contain logged information 43
the SYSOUT class for 43
starting automatically 44
terminating for interface messages 43
logical database 6, 15
Logical Unit 6.2 (LU 6.2) 2
logical unit of work, see LUW 10
LOGLEVEL configuration parameters 45, 61
LOGLEVEL parameter 50
LOGMODE parameter 50
LOGPOOL parameter 50
LTMBs, specifying the primary and secondary number of 50
LU
defined 8
displaying for each service address 28
name, specifying in the gateway configuration member 57
names, mapping to TP names 50
specifying the name of a remote 8
supporting sessions with multiple LUs 8
LU 6.2 protocol 2, 8
LU-LU session 8
LUW parameter, used with THREAD keyword to determine opening and closing of DB2 thread 56
LUW, DB2 interface software opening a thread when LUW begins 10

M

M severity flag 45, 50, 61
macro_name variable, used with SQLMACRO 59
MANAGER operand
used with !SNAP 33
used with !TRACE 42
Master Console Information 50
Master Control task
activating tracing in the 42
capturing the data areas for 33
MASTER operand
used with !SNAP 33
used with !TRACE 42
master scheduler address space, validating that the caller is 50
MAXECSA
exceeding 56
exceeding with allocation of secondary cell pool 50
parameter 50
MAXSERVICES parameter 50
members, samples for configuration 49
message severity level
E 50
I 50
S 50
W 50
MESSAGE-CASE parameter 50

messages
avoiding display with NOPROMPT 39
controlling logging of 43
generated by Host Services subsystem 46, 62
interpreting 45, 61
severity flag 45, 46, 61, 62
specifying which interfaces receive 50
written to Host Services log 50
Messages and Codes manual 45, 61
met.CONFIG data set, requiring a separate gateway configuration member in 49
METCAS cataloged procedure, default name of the cataloged procedure used to start a CAS service 50
METCNCTX, specifying the name of the exit routine in the Control member 50
METCNTRL 35
default member name for Control member in SYSPARM data set 49
holding default values for all configuration parameters 49
METDBA, sample DB2 authorization ID name 13
METPLANC 50
METPLANR 50
METSERV operand
used with !START 35
used with !STOP 39
METSQLX 50
METSQS default Control member parameter 50
METSTATR 50
METSVC routine 56
MIXED case interface and log messages 50
MONITOR operand
used with !SNAP 33
used with !TRACE 42
monitor process 4
Monitor task
activating tracing in the 42
capturing the data areas for 33
specifying the application ID name for APPC conversations for 50
specifying the remote LU name and PU type for sending 'Service started' and 'Service stopped' messages 57
monitoring sessions 3
MVS host interface, interpreting messages 45, 61

N

network addressable unit 8
network node 7
node, SNA network 7
NOPROMPT operand, used with !STOP 39

O

OCIBs, specifying the number for each allocation of a cell pool in ECSA 50
OPERATOR operand
used with !SNAP 33
used with !TRACE 42
Operator task
activating tracing in the 42
capturing the data areas for 33
operator-command input blocks, see OCIBs 50
OPERPOOL parameter 50
OPTIONS operand, used with !DISPLAY 28
OS/2 Database Gateway Service, one of several IDS database gateway service types 2

P

pacing levels 8
parallel sessions 8
parm_1 variable, used with SQLMACRO 59
parm_n variable, used with SQLMACRO 59
PARTNER-LU-FOR-GTH parameter 57
PARTNER-LU-FOR-HTG parameter 57
Password field 20
password, authenticating 50
PERFORM parameter 50, 57
performance groups, specifying 50, 57
PERM-CONNECT parameter 50
permanent connections, specifying 50
physical database 6, 15
PLAN_TABLE table 21
primary subsystem address space, validating that the caller is 50
privileges minimum required for
existing application plan 14
new application plan 14
problem determination, writing trace information to a central file 41
PU type, specifying in the gateway configuration member 57

Q

QMF 2

R

RACROUTE macro, using for user authentication 50
recoverable error 46, 62
recoverable error message severity level 50
remote LU name, specifying in the gateway configuration member 57
REORG
  minimum privilege required 17
  utility, running 17
REQUIRE-DPF parameter 50
resource consumption, minimizing 50
restarting a connection 4
REXX
  CAS allows users to work with 2
  commands executed through TSO/E 6
  executing in the same CAS service address space as TSO/E 10
  procedures invoking cooperative application programs 7
ROUTCDE parameter 50
RUNSTATS utility, running 17

S

S severity flag 46, 50, 62
sample exit routine METSQLX 50
SDLC, connection between host and database gateway service 2
SDSF, preventing users from entering TSO commands through 50
Secured field 20
security 14
  defining user IDs to the host 13
  level, specifying the minimum in the Control member 50
  using DPF 50
security parameters 8
SELECT privileges, DBA must have for DB2 tables 13
service address space
  identifier number 50
  introduced 2
  LU lines in a gateway configuration member 57
  naming convention for communicating with a DB2 gateway 50
  specifying permanent connection for 50
service address spaces
  activating tracing in 42
  capturing
    data areas for 34
    SGDA for 34
service address spaces *(continued)*
  communications path into 9, 10
  disabling 25
  displaying
    current status for 28
    gateway identifier connected to 28
    hexadecimal address space identifier for 28
    information for SQL address spaces 29
    service LU name for 28
    service name for 28
  enabling 30, 31
  exchanging data with a database gateway service 9, 10
  identifying source of message 46, 62
  one set for each gateway 9
  separate set for each gateway service 11
  shutting down all 39
  specifying maximum that service manager can start 50
  starting 38
  stopping
    all 39
    all controlled by a gateway 41
    communications for 41
    for a gateway 40
  terminating
    for a gateway 49
    prior to starting a new Service Manager 35
  using one for each connection to a CAS transformer 10
  using one for each connection to a data access tool 9
SERVICE ALL operands, used with !SNAP 33
service ID 28
service identifiers, displaying the list of 31
service logical unit name, displaying for a service address space 28
SERVICE LU(S) parameter 57
service manager
  commands, in Control configuration member 50
  configuration member 24
  tasks
    activating tracing in the 42
    capturing the data areas for all 33
service manager address space
  command summary for 23
  communications path into 11
  communications path to the database gateway service 11
  configuration information for 49

service manager address space *(continued)*
controlling service address spaces with 11
including a SYSLOG DD statement in the cataloged procedure for 44
initializing 49
introduced 9
purpose of 11
shutting down all subtasks for 39
starting 35
stopping 39
service manager, function in maintaining CAS sessions 6
service name, displaying for each address space 28
service request blocks, see SRBs 11
SERVICE service_name operands, used with !SNAP 34
service_name variable
used with !DISPLAY 28
used with !STOP 39
used with !TRACE 42
service_number variable
used with !DISABLE 26, 28
used with !ENABLE 31
used with !START 38
used with !STOP 41
SERVICES operand, used with !TRACE 42
session
defined 8
limits 8
mode name 8
parallel 8
supporting more than one concurrently 8
session process 4
sessions
establishing through Communications Manager/2 4
running multiple 4
SET CURRENT DEGREE='ANY' 50
severe error 46, 62
severe error message severity level 50
severity flag for messages 45
severity flag for messages written to host interface 50, 61
SGDA ALL operands, used with !SNAP 34
SGDA service_name operands, used with !SNAP 34
SGDA, capturing for service address spaces 34
SNA APPC communications, understanding 7
SNA network node 7
SNA, connection between host and database gateway service 2
SPUFI 14
SQL gateway configuration member 49
SQL ID field 20
SQL service address spaces
activating tracing in 42
capturing global and local data areas for 34
function 9
introduced 2, 9
SQL statements, setting each time the SQL service opens a thread 50
SQL/DS, accessing data through DRDA 3
SQLEXIT parameter, specified in the Control member 50
SQLMACRO macro_name 59
SQLMACRO value_1 59
SQLMACRO value_n 59
SQLMACRO facility, using 59
SQLMACRO parm_1 59
SQLMACRO parm_n 59
SQS operand
used with !DISPLAY 29
used with !SNAP 34
used with !TRACE 42
SQSMEM parameter 50
SQSPROC parameter 50, 57
SRBPOOL parameter 50
SRBs
specified by SRBPOOL parameter for each allocation of a cell pool in ECSA 50
specifying for each allocation of a cell pool in ECSA 50
used to carry operator commands to service address spaces 11
SSDA operand, used with !SNAP 34
SSDA, capturing with the !SNAP command 34
START operand, used with !TRACE 42
starting a service address space 50
STATS, minimum privilege required 17
status messages, interpreting 45, 61
status reports, specifying seconds to wait between 50
STATUS, displaying for a service address space 28
STATUSEXIT parameter 50
STATUSINTERVAL parameter 50
STEPLIBS, for testing subsystems or gateways 50, 57
STOGROUPS, creating to be used by the database 16
STOP operand
used with !SNAP 34
used with !TRACE 43
used with !WRITELOG 43

stopping host services for a specified gateway 39
storage
  specifying dynamic storage for CAS data encoding/decoding 50
  specifying dynamic storage for SQL data encoding/decoding 50
subsystem name, specifying the default for DB2 connections 50
subsystem specification (IEFSSNxx), SVC number named in 56
subtask, ATTACH 50
support
  technical, for IDS 46
support, technical for IDS 62
SVC
  number, specifying for Host Services routines to use 56
  testing a new 56
  validating calls with 50
SVC-CALLER-VERIFICATION parameter 50
SVCNUMBER parameter 56
synonyms 18
synonyms, creating with macros 21
syntax conventions 24
SYS1.LPALIB, METSVC routine installed in 56
SYSIBM.SYSCOLUMNS 13
SYSIBM.SYSINDEXES 13
SYSIBM.SYSKEYS 13
SYSIBM.SYSSYNONYMS 13
SYSIBM.SYSTABLES 13
SYSLOG DD statement 44
SYSOUT class
  operands
    used with !SNAP 34
    used with !WRITELOG 43
  specifying for allocating trace data sets 56
  specifying for captured information 34
  specifying for logged information 43
SYSPARM 49
System Display and Search Facility, see SDSF 50
SYSTEM tables
  creating synonyms for 21

T

T severity flag 46, 50, 61
table descriptions, loading into the WTDD 16
table privileges, granting 14
table space, creating 16
tables, conditions for user access 20
tablespace, for WTDD, created by CREATEDB job 15
TANBs, specifying for each allocation of a cell pool in ECSA 56
TANPOOL parameter 56
task
  identifying message source 46, 62
  Monitor 50
task action/notification blocks, see TANBs 56
technical support, contacting 46, 62
terminating
  active service address spaces 35
  service manager address space 39
THREAD parameter 56
timing out 50
token ring, used by IDS database gateway service 5
TP names
  associated with a service's APPC conversation 49
  mapping to LU names 50
  setting the limit in the gateway configuration member 50
  specifying the prefix 57
TP, defined 7
trace
  starting a 42
  terminating a 43
trace information, writing to a central file 41
trace message 46, 50, 61
trace table, specifying number of entries to allocate per task 50
TRACE-CLASS parameter 56
tracing
  activating all 42
  experiencing difficulties with 50
  messages, local to service address space 11
TSO/E 6, 10
type, gateway 57

U

UPPER case interface and log messages 50
USE OF BPO WITH GRANT OPTION, required if storage groups to be used by the database already exist 16
user ID, authenticating 50
User Name field 20
utilities, running against the WTDD tablespace 17

V

value variable, used with !SET 32
value_1 variable, used with SQLMACRO 59
value_n variable, used with SQLMACRO 59
value, assigning to a configuration parameter 32
VTAM
  establishing SNA sessions and APPC conversations through 4
  specifying the logon-mode table entry name 50

W

W severity flag 46, 50, 62
warning message 46, 50, 62
warning message severity level 50
watchdog process 4, 6
workstation tools, using to send SQL statements 5
writing trace information to a file 41
WTDD
  defined 15
  installing 15
  loading a self description 16
  loading with initial DB2 catalog information 15
  purpose of 6
  steps for installing 16
  tables owner 15
WTDD Owner field 20

DSMBEG323I STARTING PASS 2 OF 4.
DSMBEG323I STARTING PASS 3 OF 4.
DSMBEG323I STARTING PASS 4 OF 4.
DSMEXG708E TEXT EXCEEDS RIGHT PAGE BOUNDARY ON PAGE 29.
DSMEXG708E TEXT EXCEEDS RIGHT PAGE BOUNDARY ON PAGE 30.
DSMEXG708E TEXT EXCEEDS RIGHT PAGE BOUNDARY ON PAGE 37.
DSMEXG708E TEXT EXCEEDS RIGHT PAGE BOUNDARY ON PAGE 38.